1/5/1 (Item 1 from file: 351)
DIALOG(R)File 351:Derwent WPI


012874383 **Image available**
WPI Acc No: 2000-046216/ 200004
XRPX Acc No: N00-035791
Data decoding method in electronic commercial transaction system - involves decoding data attached with key recovery demand, using searched disclosure key of one of disclosure key certificate
Patent Assignee: HITACHI LTD (HITA )
Number of Countries: 001 Number of Patents: 001
Patent Family:

| Patent No | Kind | Date | Applicat No | Kind | Date | Week |
|---|---|---|---|---|---|---|
| JP 11308213 | A | 19991105 | JP 98109450 | A | 19980420 | 200004 B |

Priority Applications (No Type Date): JP 98109450 A 19980420
Patent Details:

| Patent No | Kind | Lan | Pg | Main IPC | Filing Notes |
|---|---|---|---|---|---|
| JP 11308213 | A | | 19 | H04L-009/32 | |

Abstract (Basic): JP 11308213 A

NOVELTY - The user is authenticated, using specific disclosure key certificate received along with key recovery demand. Then, the disclosure key of another disclosure key certificate stored in memory, is searched using two secret keys. The data attached with the recovery demand are decoded, using searched disclosure key. DETAILED DESCRIPTION - A specific public-presentation key certificate is canceled, according to the demand of user. Then, a new public presentation key certificate is published, when the demand is made from the user. The key recovery demand including the specific disclosure key certificate and the data encoded using disclosure key of another disclosure key certificate. An INDEPENDENT CLAIM is also included for data decoding apparatus used in electronic commercial transaction system.

USE - For decoding data in electronic commercial transac- tion system.

ADVANTAGE - Since management of disclosure key certificate and key decoding are performed using single unit, a pair of secret keys are made available for only the user to recognize the encoded data when disclosure key of disclosure key certificate is lost. DESCRIPTION OF DRAWING(S) - The figure shows schematic block diagram of a system for which the data decoding method is adopted.

Dwg.1/10

Title Terms: DATA; DECODE; METHOD; ELECTRONIC; COMMERCIAL; TRANSACTION; SYSTEM; DECODE; DATA; ATTACH; KEY; RECOVER; DEMAND; SEARCH; DISCLOSE; KEY ; ONE; DISCLOSE; KEY; CERTIFY
Derwent Class: P85; W01
International Patent Class (Main): H04L-009/32
International Patent Class (Additional): G09C-001/00; H04L-009/08
File Segment: EPI; EngPI

06366602 **Image available**

# ENCRYPTION DATA RECOVERY METHOD AND ITS SYSTEM

PUB. NO.: 11-308213 A]
PUBLISHED: November 05, 1999 (19991105)
INVENTOR(s): DOMYO SEIICHI
UMEKI HISASHI
TSUCHIYA HIROYOSHI
KAWAI TORU
YANAGIUCHI HIDETAKA
APPLICANT(s): HITACHI LTD
APPL. NO.: 10-109450 [JP 98109450]
FILED: April 20, 1998 (19980420)
INTL CLASS: H04L-009/32; G09C-001/00; H04L-009/08

## ABSTRACT

PROBLEM TO BE SOLVED: To allow the system to approve data recovery by using a secret key only when a user receiving an issued public key certificate loses the secret key in pairs with an open key in the certificate by unifying the user management of a public key certificate and key recovery. SOLUTION: A certification authority CA 220 abolishes a 1st public key certificate on request of a user device USC 200 and issues a 2nd public key certificate. Furthermore, a data recovery device DRC 240 authenticates the user by using the 2nd public key certificate attached to a key recovery request according to the key recovery request of the USC 200 and uses a secret key in pairs with the public key of the 1st public key certificate deposited by a key deposit device KEC 230 to decode the data encrypted by the open key of the 1st public key certificate.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平11－308213

(43)公開日　平成11年(1999)11月5日

| (51)Int.Cl.[6] | | 識別記号 | FI | | |
|---|---|---|---|---|---|
| H04L | 9/32 | | H04L | 9/00 | 675D |
| G09C | 1/00 | 630 | G09C | 1/00 | 630F |
| H04L | 9/08 | | H04L | 9/00 | 601F |

審査請求　未請求　請求項の数10　ＯＬ　（全 19 頁）

(21)出願番号　特願平10－109450

(22)出願日　平成10年(1998)４月20日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　道明　誠一
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　梅木　久志
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　土屋　宏嘉
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内

(74)代理人　弁理士　富田　和子

最終頁に続く

(54)【発明の名称】　暗号データ回復方法および装置

(57)【要約】

【課題】公開鍵証明書および鍵回復の利用者管理を１元化し、公開鍵証明書の発行を受けた利用者が当該証明書の公開鍵と対の秘密鍵を紛失した場合にのみ、当該秘密鍵を用いたデータ回復を認めるようにする。

【解決手段】ＣＡ２２０は、ＵＳＣ２００の要求にしたがい第１の公開鍵証明書を廃止して、第２の公開鍵証明書を発行する。また、ＤＲＣ２４０は、ＵＳＣ２００の鍵回復要求にしたがい、当該鍵回復要求に付された第２の公開鍵証明書を用いて認証した後、ＫＥＣ２３０に供託された第１の公開鍵証明書の公開鍵と対の秘密鍵を用いて、前記第１の公開鍵証明書の公開鍵で暗号化されたデータを復号する。

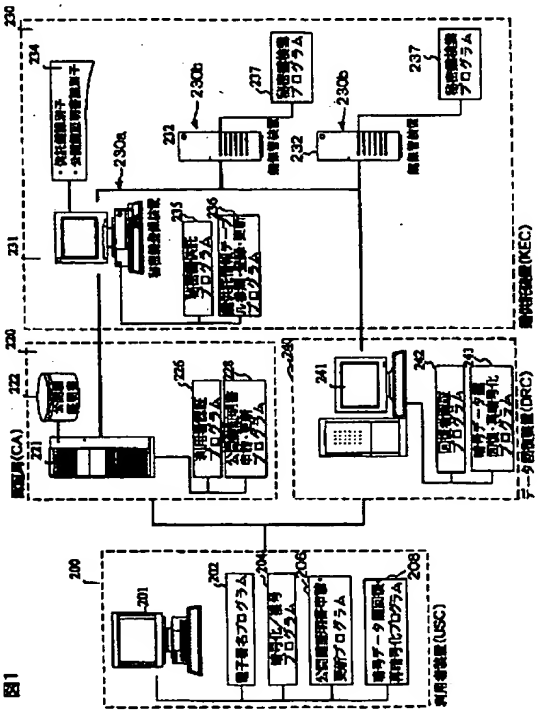


図1

1

【特許請求の範囲】

【請求項１】認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵により暗号化されたデータを回復する暗号データ回復方法であって、

認証局において、

利用者の要求にしたがい、第１の公開鍵証明書を発行する第１のステップと、

前記利用者の要求にしたがい、当該第１の公開鍵証明書を廃止して、第２の公開鍵証明書を発行する第２のステップと、

前記利用者からの、前記第１の公開鍵証明書の公開鍵により暗号化されたデータおよび前記第２の公開鍵証明書を用いた鍵回復要求を受け付ける第３のステップと、

前記第３のステップで受け付けた鍵回復要求にしたがい、当該鍵回復要求に付された前記第２の公開鍵証明書を用いて、前記利用者を認証する第４のステップと、

前記第４のステップにより、前記利用者が認証された場合に、前記第１の公開鍵証明書の公開鍵と対の秘密鍵を、供託されている鍵の中から検索する第５のステップと、

前記第５のステップで検索した秘密鍵を用いて、前記鍵回復要求に付された前記暗号化されたデータを回復する第６のステップと、を備えることを特徴とする暗号データ回復方法。

【請求項２】請求項１記載の暗号データ回復方法であって、

前記第５のステップは、前記第１の公開鍵証明書が前記利用者に複数発行されている場合、前記複数の第１の公開鍵証明書各々の有効期限と、前記鍵回復要求に付された前記暗号化されたデータの暗号化時刻とを比較して、当該暗号化されたデータを暗号化するのに用いた秘密鍵を、供託されている鍵の中から検索することを特徴とする暗号データ回復方法。

【請求項３】認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵により暗号化されたデータを、当該秘密鍵を供託された鍵回復センタにより回復する暗号データ回復方法であって、

認証局において、

利用者の要求にしたがい、第１の公開鍵証明書を発行する第１のステップと、

前記利用者の要求にしたがい、当該第１の公開鍵証明書を廃止して、第２の公開鍵証明書を発行する第２のステップと、

前記第２のステップで廃止した前記第１の公開鍵証明書の公開鍵と対の秘密鍵を示す供託鍵識別子を、当該利用者に送付する第３のステップと、

鍵回復センタにおいて、

前記利用者からの、前記第１の公開鍵証明書の公開鍵により暗号化されたデータと前記供託鍵識別子とが連結されたデータに、前記第２の公開鍵証明書の公開鍵と対の

2

秘密鍵による署名が付されたデータを、鍵回復要求として受け付ける第４のステップと、

前記第４のステップで受け付けた鍵回復要求に付された前記署名を、前記第２の公開鍵証明書の公開鍵を用いて検証することで、前記利用者を認証する第５のステップと、

前記第５のステップにより、前記利用者が認証された場合に、前記第４のステップで受け付けた鍵回復要求に付された供託鍵識別子が示す秘密鍵を、供託されている鍵の中から検索する第６のステップと、

前記第６のステップで検索した秘密鍵を用いて、前記鍵回復要求に付された、前記暗号化されたデータを回復する第７のステップと、を備えることを特徴とする暗号データ回復方法。

【請求項４】請求項３記載の暗号データ回復方法であって、

鍵回復センタにおいて、

前記第７のステップは、回復したデータを、前記第２の公開鍵証明書の公開鍵を用いて再暗号化して利用者に提供することを特徴とする暗号データ回復方法。

【請求項５】請求項３記載の暗号データ回復方法であって、

認証局において、

前記第２のステップは、廃止した第１の公開鍵証明書の公開鍵と対の秘密鍵を用いた鍵回復の有効期限を設定し、

鍵回復センタにおいて、

前記第５のステップは、前記第４のステップで受け付けた鍵回復要求が、前記２のステップで設定した有効期限を過ぎている場合、前記認証処理を行わないことを特徴とする暗号データ回復方法。

【請求項６】請求項３記載の暗号データ回復方法であって、

認証局において、

前記第３のステップは、前記第２のステップで廃止した前記第１の公開鍵証明書の公開鍵と対の秘密鍵を示す供託鍵識別子を、当該供託鍵識別子に有効期限を設定して利用者に送付し、

鍵回復センタにおいて、

前記第５のステップは、前記第４のステップで受け付けた鍵回復要求に付された前記暗号化されたデータの暗号化時刻が、当該鍵回復要求に付された供託鍵識別子の有効期限を過ぎている場合、前記認証処理を行わないことを特徴とする暗号データ回復方法。

【請求項７】認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵により暗号化されたデータを回復する暗号データ回復システムであって、

鍵供託装置と、利用者装置と、データ回復装置とを備え、

前記鍵供託装置は、

認証局が廃止した第１の公開鍵証明書の公開鍵と対の秘密鍵を保管する鍵保管手段と、

前記データ回復装置の指示にしたがい、当該装置から通知された供託鍵識別子が示す秘密鍵を前記鍵保管手段から検索し、前記データ回復装置に通知する通知手段と、を有し、

前記利用者装置は、

前記第１の公開鍵証明書の公開鍵で暗号化されたデータと、認証局から送付された前記第１の公開鍵証明書の公開鍵と対の秘密鍵を示す供託鍵識別子とを連結し、前記連結したデータに対して、認証局が発行した前記第２の公開鍵証明書の公開鍵と対の秘密鍵により電子署名を生成し、当該電子署名がなされた前記連結したデータを、前記データ回復装置に通知する電子署名生成手段を有し、

前記データ回復装置は、

前記利用者装置から通知された前記電子署名を、前記第２の公開鍵証明書の公開鍵を用いて検証することで、利用者を認証する認証手段と、

前記認証手段により、利用者が認証された場合に、前記利用者装置から通知された前記連結したデータを構成する供託鍵識別子を、前記鍵供託装置に通知して、当該供託鍵識別子が示す秘密鍵を取得する取得手段と、

前記取得手段により取得した秘密鍵を用いて、前記利用者装置から通知された前記連結したデータを構成する暗号化されたデータを回復する回復手段と、を有することを特徴とする暗号データ回復システム。

【請求項８】認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵により暗号化されたデータを回復する暗号データ回復装置であって、

認証局が廃止した第１の公開鍵証明書の公開鍵と対の秘密鍵を保管する鍵保管手段と、

前記第２の公開鍵証明書の公開鍵と対の秘密鍵により電子署名がなされた、前記第１の公開鍵証明書の公開鍵で暗号化されたデータと前記第１の公開鍵証明書の公開鍵と対の秘密鍵を示す供託鍵識別子とが連結されたデータを、利用者から受け取る受信手段と、

前記第２の公開鍵証明書の公開鍵を用いて、前記連結されたデータに付された電子署名を検証することで、利用者を認証する認証手段と、

前記認証手段により、利用者が認証された場合に、前記受信手段により受信した前記連結されたデータを構成する供託鍵識別子が示す秘密鍵を、前記鍵保管手段から取得する取得手段と、

前記取得手段により取得した秘密鍵を用いて、前記受信手段により受信した前記連結されたデータを構成する暗号化されたデータを回復する回復手段と、を備えていることを特徴とする暗号データ回復装置。

【請求項９】認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵により暗号化されたデータを回復させるため



のプログラムが記憶された記憶媒体であって、

当該プログラムは、情報処理装置に、

認証局が廃止した第１の公開鍵証明書の公開鍵と対の秘密鍵を記憶装置に保管する第１のステップと、

前記第２の公開鍵証明書の公開鍵と対の秘密鍵により電子署名がなされた、前記第１の公開鍵証明書の公開鍵で暗号化されたデータと前記第１の公開鍵証明書の公開鍵と対の秘密鍵を示す供託鍵識別子とが連結されたデータを、利用者から受け取る第２のステップと、

前記第２の公開鍵証明書の公開鍵を用いて、前記第２のステップにより受信した前記連結されたデータに付された電子署名を検証することで、利用者を認証する第３のステップと、

前記第３のステップにより、利用者が認証された場合に、前記第２のステップにより受信した前記連結されたデータを構成する供託鍵識別子が示す秘密鍵を、記憶装置から取得する第４のステップと、

前記第４のステップにより取得した秘密鍵を用いて、前記第２のステップにより受信した前記連結されたデータを構成する暗号化されたデータを回復する第５のステップと、を実行させることを特徴とするプログラムが記憶された記憶媒体。

【請求項１０】送信者の装置において、

取引データを、認証局が発行した第１の公開鍵証明書の公開鍵を用いて暗号化し、受信者に送信するステップと、

受信者の装置において、

前記送信者から送信された前記暗号化された取引データを記憶媒体に保管するステップと、

前記記憶媒体に保管した前記暗号化された取引データを、前記第１公開鍵証明書の公開鍵と対の秘密鍵を用いて復号するステップと、を備える電子商取引方法であって、

認証局において、

前記受信者の要求にしたがい、当該第１の公開鍵証明書を廃止して、第２の公開鍵証明書を発行する第１のステップと、

前記受信者からの、前記暗号化された取引データおよび前記第２の公開鍵証明書を用いた鍵回復要求を受け付ける第２のステップと、

前記第２のステップで受け付けた鍵回復要求にしたがい、当該鍵回復要求に付された前記第２の公開鍵証明書を用いて、前記受信者を認証する第３のステップと、

前記第３のステップにより、前記受信者が認証された場合に、前記第１の公開鍵証明書の公開鍵と対の秘密鍵を、供託されている鍵の中から検索する第４のステップと、

前記第４のステップで検索した秘密鍵を用いて、前記鍵回復要求に付された前記暗号化された取引データを回復する第５のステップと、を備えることを特徴とする電子

商取引方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、利用者の公開鍵で暗号化されたデータを、当該利用者の要求に応じて復号化する暗号データ回復技術、特に、エンベロープ（Envelope）型暗号システムにおいて、利用者の公開鍵でカプセル化されたセッション鍵を、当該利用者の要求に応じてカプセル開放する鍵回復技術に関する。

【０００２】

【従来の技術】暗号技術は、情報の盗聴や改ざん、あるいは利用者のなりすましなどを防止する計算機技術である。データを暗号化する効果の一つとして、情報にアクセス(復号)できるのは、復号用の鍵を持つ利用者に限定されることを挙げることができる。データの暗号化は、計算機システムの運用(オン/オフライン)やプラットフォーム（ＯＳ）に依存することなく、機密情報のアクセス制御を容易に実現する手段といえる。

【０００３】機密情報の保管に関し、通常、以下の条件が課せられる。

【０００４】（１）法律や組織の規定によって長期間(たとえば年単位)保存の義務がある。

【０００５】（２）いったん保存したら、ほとんどアクセスすることがない。

【０００６】（３）外部あるいは社内の監査時には、確実に提出（復号）できるものでなければならない。

【０００７】したがって、機密情報を提出しようとしたが復号できない場合は、組織や担当部署の不利益となることが予想される。復号できない理由としては、保存したはずの機密情報が見つからない場合の他に、復号用の鍵を紛失した場合や、担当者の変更により鍵の所在がわからなくなった場合など、鍵管理に問題がある場合が考えられる。

【０００８】従来、暗号システムの鍵管理は、利用者やシステム運用者(とくに機密情報の保存責任者)による自主的な管理に任せている場合が多い。したがって、機密情報の暗号化がさらに普及するにつれ、鍵管理をめぐるトラブルが多発することが予想される。

【０００９】そこで、復号用の鍵を、利用者ではなくシステムで一元管理し、いざというときに取出すバックアップシステムが提案されている。この技術は、鍵回復技術と呼ばれている。とくに、鍵回復技術のうち、利用者の秘密鍵をシステムに供託するタイプの方式は、鍵供託方式と呼ばれている。

【００１０】鍵供託方式では、利用者端末、鍵供託装置(鍵保管装置)、およびデータ回復装置など、機能ごとに専門の計算機装置を配置する分散システムを構築している。このような鍵回復方式の公知例として、たとえば「暗号データ回復方法、鍵登録システムおよびデータ回復システム(特願平９－８００８１号）」がある。



【００１１】また、公開鍵暗号アルゴリズムを用いたシステムでは、秘密鍵の安全管理、および公開鍵の信頼性確保を図るため、公開鍵証明書を発行する専門機関である認証局を設立し、当該認証局が発行した公開鍵証明書を用いて認証を行う利用者認証技術が提案されている。当該技術では、利用者は、認証局ならびに公開鍵証明書を信頼することで、相手を認証したり、メッセージの完全性を確認したりすることができる。このような利用者認証方式の公知例として、たとえば「暗号通信における公開鍵登録方法および公開鍵証明書の発行局（特開平７－１６０１９８）」がある。

【００１２】

【発明が解決しようとする課題】ところで、上記説明した従来の鍵回復方式および利用者認証方式には、以下のような問題がある。

【００１３】従来は、鍵の供託および供託された鍵を用いてデータの回復を行う機関（鍵回復システム）と、公開鍵証明書を発行する認証局とが別個独立して設けられている。このため、公開鍵証明書により証明された公開鍵と対の秘密鍵を供託しようとする場合、利用者は、まず認証局に公開鍵証明書を発行してもらい、次に、当該公開鍵証明書の公開鍵と対の秘密鍵を鍵回復システムに供託する。つまり、利用者は、認証局および鍵回復システムに対して、別個独立に手続きを行うことになる。

【００１４】したがって、認証局および鍵回復システムは、各々の利用者を互いに無関係に管理することになる（２元管理）。このため、鍵回復システムに供託した、認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵を紛失（ここでいう紛失とは、秘密鍵が格納されたＩＣカードを物理的に紛失した場合の他に、秘密鍵を格納した鍵ファイルをオープンするパスワードを忘れる場合など、事実上、秘密鍵が使用不可能な状態も含む）した場合、利用者は、鍵回復センタに対して、紛失した秘密鍵を用いた暗号データの復号を依頼することになる。この場合、認証局からすれば、利用者の管理を鍵回復システムとは別個独立して行っているので、利用者からその事実（認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵を紛失したこと）が伝えられない限り、把握することができない。

【００１５】また、紛失した秘密鍵を用いた暗号データの復号を依頼する利用者が、当該秘密鍵と対になる公開鍵の公開鍵証明書の発行を受けた者であるという保証もない。つまり、認証局および鍵回復センタ各々に登録された利用者が同一人物であるという保証がない。

【００１６】本発明は、上記事情に鑑みてなされたものであり、認証局および鍵回復システムにおける利用者管理を１元化し、認証局による公開鍵証明書の発行を受けた利用者に対してのみ、当該公開鍵証明書の公開鍵と対の秘密鍵を紛失した場合に、当該秘密鍵を用いたデータ回復を認めるデータ回復方法を提供することにある。

【００１７】
【課題を解決するための手段】上記課題を解決するために、本発明の第１の態様は、認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵により暗号化されたデータを回復する暗号データ回復方法であって、認証局において、利用者の要求にしたがい、第１の公開鍵証明書を発行する第１のステップと、前記利用者の要求にしたがい、当該第１の公開鍵証明書を廃止して、第２の公開鍵証明書を発行する第２のステップと、前記利用者からの、前記第１の公開鍵証明書の公開鍵により暗号化されたデータおよび前記第２の公開鍵証明書を用いた鍵回復要求を受け付ける第３のステップと、前記第３のステップで受け付けた鍵回復要求にしたがい、当該鍵回復要求に付された前記第２の公開鍵証明書を用いて、前記利用者を認証する第４のステップと、前記第４のステップにより、前記利用者が認証された場合に、前記第１の公開鍵証明書の公開鍵と対の秘密鍵を、供託されている鍵の中から検索する第５のステップと、前記第５のステップで検索した秘密鍵を用いて、前記鍵回復要求に付された前記暗号化されたデータを回復する第６のステップと、を備えることを特徴とする。
【００１８】上記第１の態様によれば、認証局は、第１の公開鍵証明書（データ暗復号用）を発行した利用者が当該証明書の公開鍵と対の秘密鍵を紛失した場合、当該利用者から要求があった場合にのみ、当該証明書を廃止して、前記利用者に第２の公開鍵証明書（認証用）を発行するようにしている。
【００１９】そして、前記第２の公開鍵証明書を用いて利用者を認証することにより、前記第２の公開鍵証明書を発行を受けた利用者にのみ、第１の公開鍵証明書の公開鍵と対の秘密鍵を用いたデータ復号（たとえば、セッション鍵のカプセル開放）を認めるようにしている。
【００２０】このようにすることで、鍵回復を公開鍵証明書の利用者にのみ認めることになり、このため、公開鍵証明書の利用者の認証・登録手段と鍵回復の利用者の認証・登録手段とを共通化することができるので、利用者管理を１元化することができる。また、認証局による公開鍵証明書の発行を受けた利用者に対してのみ、当該公開鍵証明書の公開鍵と対の秘密鍵を紛失した場合に、当該秘密鍵を用いたデータ回復を認めることができる。
【００２１】なお、上記第１の態様において、前記第５のステップは、前記第１の公開鍵証明書が前記利用者に複数発行されている場合、前記複数の第１の公開鍵証明書各々の有効期限と、前記鍵回復要求に付された前記暗号化されたデータの暗号化時刻とを比較して、当該暗号化されたデータを暗号化するのに用いた秘密鍵を検索するものでもよい。
【００２２】このようにすることで、たとえば、利用者が、認証局からデータ暗復号用の第１の公開鍵証明書を複数発行してもらっている場合において、相手から、そのうちの１つの公開鍵を用いて暗号化されたデータを受け取った場合、対応する秘密鍵が分からなくなってしまいデータを復号できない場合でも、認証局に鍵回復要求をすることで、当該データの回復を行うことができる。
【００２３】また、本発明の第２の態様は、認証局が発行した公開鍵証明書の公開鍵と対の秘密鍵により暗号化されたデータを、当該秘密鍵を供託された鍵回復センタにより回復する暗号データ回復方法であって、認証局において、利用者の要求にしたがい、第１の公開鍵証明書を発行する第１のステップと、前記利用者の要求にしたがい、当該第１の公開鍵証明書を廃止して、第２の公開鍵証明書を発行する第２のステップと、前記第２のステップで廃止した前記第１の公開鍵証明書の公開鍵と対の秘密鍵を示す供託鍵識別子を、当該利用者に送付する第３のステップと、鍵回復センタにおいて、前記利用者からの、前記第１の公開鍵証明書の公開鍵により暗号化されたデータと前記供託鍵識別子とが連結されたデータに、前記第２の公開鍵証明書の公開鍵と対の秘密鍵による署名が付されたデータを、鍵回復要求として受け付ける第４のステップと、前記第４のステップで受け付けた鍵回復要求に付された前記署名を、前記第２の公開鍵証明書の公開鍵を用いて検証することで、前記利用者を認証する第５のステップと、前記第５のステップにより、前記利用者が認証された場合に、前記第４のステップで受け付けた鍵回復要求に付された供託鍵識別子が示す秘密鍵を、供託されている鍵の中から検索する第６のステップと、前記第６のステップで検索した秘密鍵を用いて、前記鍵回復要求に付された、前記暗号化されたデータを回復する第７のステップと、を備えることを特徴とする。
【００２４】本発明の第２の態様においても、第１の態様と同様、公開鍵証明書の利用者の認証・登録手段と鍵回復の利用者の認証・登録手段とを共通化することができるので、利用者管理を１元化することができる。また、認証局による公開鍵証明書の発行を受けた利用者に対してのみ、当該公開鍵証明書の公開鍵と対の秘密鍵を紛失した場合に、当該秘密鍵を用いたデータ回復を認めることができる。
【００２５】なお、上記第２の態様において、前記第２の公開鍵証明書を利用者認証用兼データ暗号化用として用い、前記第７のステップにおいて、回復したデータを、前記第２の公開鍵証明書の公開鍵を用いて再暗号化して利用者に提供するようにしてもよい。
【００２６】このようにすることで、回復したデータを利用者に通知する際のセキュリティを向上させることができる。
【００２７】また、前記第２のステップにおいて、廃止した第１の公開鍵証明書の公開鍵と対の秘密鍵を用いた鍵回復の有効期限を設定するようにし、前記第５のステップにおいて、前記第４のステップで受け付けた鍵回復

要求が前記２のステップで設定した有効期限を過ぎている場合、前記データ回復処理を行わないようにしてもよい。

【００２８】あるいは、前記第３のステップにおいて、前記第２のステップで廃止した前記第１の公開鍵証明書の公開鍵と対の秘密鍵を示す供託鍵識別子を、当該供託鍵識別子に有効期限を設定して利用者に送付するようにし、前記第５のステップにおいて、前記第４のステップで受け付けた鍵回復要求に付された前記暗号化されたデータの暗号化時刻が当該鍵回復要求に付された供託鍵識別子の有効期限を過ぎている場合、前記データ回復処理を行わないようにしてもよい。

【００２９】このようにすることで、鍵回復およびデータ回復を、第２の公開鍵証明書の発行（つまり、第１の公開鍵証明書の廃止）から所定期間にのみ認めるよう限定することができる。

【００３０】

【発明の実施の形態】以下に、本発明の第１実施形態について説明する。

【００３１】図１は、本発明の第１実施形態が適用されたシステムの概略構成図である。

【００３２】本実施形態が適用されたシステムは、図１に示すように、利用者装置（ＵＳＣ：User Security Component）２００と、ＵＳＣ２００からの要求に応じて公開鍵証明書を発行する認証局（ＣＡ：Certificate Authority）２２０と、ＣＡ２２０が発行した公開鍵証明書の公開鍵と対の秘密鍵を供託する鍵供託装置（ＫＥＣ：Key Escrow Component）２３０と、ＵＳＣ２００からの鍵回復要求に応じてＫＥＣ２３０に供託された秘密鍵を用いて、カプセル化されたセッション鍵を開放するデータ回復装置（ＤＲＣ：Data Recovery Component）２４０と、を備えて構成されている。

【００３３】本実施形態では、ＫＥＣ２３０がＣＡ２２０のバックエンド装置として設けられている。この点、従来の認証局および鍵回復システムと異なる。なお、図１では、ＣＡ２２０、ＫＥＣ２３０、およびＤＲＣ２４０を、別個独立した装置として、分離して表示しているが、これは、従来の認証局および鍵回復システムの構成との差異を明確にするために、便宜上、そのように表示しているだけである。ＣＡ２２０、ＫＥＣ２３０、およびＤＲＣ２４０を、１つの装置上で実現することにより、鍵回復機能を有する認証局としても構わない。

【００３４】ＵＳＣ２００は、計算機装置２０１において、電子署名プログラム２０２、暗号化／復号プログラム２０４、公開鍵証明書申請・更新プログラム２０６、および暗号データ鍵回復・再暗号化２０８がメモリ上にロードされ、ＣＰＵにより実行されることで実現される。

【００３５】これらのプログラムは、たとえばＣＤ－ＲＯＭなどの記憶媒体により、計算機装置２０１に提供されるものでもよい。あるいは、ＩＣカードなどの計算機能を備えた記憶媒体に格納しておき、計算機装置２０１に当該記憶媒体が装着されると、当該記憶媒体自体がこれらのプログラムを実行するようにしてもよい。

【００３６】ＣＡ２２０は、公開鍵証明書を格納するデータベース（ＤＢ）２２２を備えた計算機装置２２１において、利用者認証プログラム２２６、および公開鍵証明書申請・更新プログラム２２８がメモリ上にロードされ、ＣＰＵにより実行されることで実現される。

【００３７】ＫＥＣ２３０は、供託の対象となる秘密鍵を登録する秘密鍵登録装置２３０ａと、秘密鍵登録装置２３０ａにより登録された秘密鍵を保管するとともに、秘密鍵登録装置２３０ａからの指示にしたがい保管した秘密鍵の検索を行う鍵保管装置２３０ｂと、で構成される。

【００３８】秘密鍵登録装置２３０ａは、計算機装置２３１において、ＣＡ２２０から送られてきた秘密鍵を、鍵保管装置２３０ｂに登録（供託）するための秘密鍵供託プログラム２３５、および、秘密鍵を供託した証拠としての付加情報（鍵供託識別子）と当該秘密鍵と対の公開鍵についての公開鍵証明書を示す識別子とが互いに関連付けられたデータを記憶する鍵供託情報テーブル２３４の参照・登録・更新処理を行う鍵供託情報テーブル参照・登録・更新プログラム２３６が、メモリ上にロードされ、ＣＰＵにより実行されることで実現される。

【００３９】鍵保管装置２３０ｂは、計算機装置２３２において、秘密鍵登録装置２３０ａからの指示にしたがい、保管した秘密鍵から所定の秘密鍵を検索するための秘密鍵検索プログラム２３７がメモリ上にロードされ、ＣＰＵにより実行されることで実現される。

【００４０】なお、秘密鍵登録装置２３０ａおよび鍵保管装置２３０ｂは、１つの装置上で実現されるようにしてもよい。

【００４１】ＤＲＣ２４０は、計算機装置２４１において、ＵＳＣ２００を用いて鍵回復要求を行った利用者の認証を行うための回復者認証プログラム２４２、および、ＵＳＣ２００から送られてきた、カプセル化されたセッション鍵を開放し、当該鍵を再暗号化する暗号データ鍵回復・再暗号化プログラム２４３が、メモリ上にロードされ、ＣＰＵにより実行されることで実現される。

【００４２】次に、以上説明したシステムの処理手順について説明する。

【００４３】本実施形態の処理手順は、利用者が、ＣＡ２２０に第１の公開鍵証明書の発行を依頼してから、当該証明書の廃止して第２の公開鍵証明書を再発行してもらうまでの処理手順と、利用者が、再発行を受けた第２の公開鍵証明書を用いて認証を受けることで、廃止した第１の公開鍵証明書の公開鍵でカプセル化されたセッション鍵をカプセル開放してもらうまでの処理手順とに、分けて考えることができる。

【００４４】まず、利用者が、ＣＡ２２０に第１の公開鍵証明書の発行を依頼してから、当該証明書の廃止して第２の公開鍵証明書を再発行してもらうまでの処理手順について、図２を参照して説明する。

【００４５】図２は、本発明の第１実施形態において、第１の公開鍵証明書を発行してから、当該証明書を廃止して第２の公開鍵証明書を再発行するまでの処理手順を説明するためのフロー図である。

【００４６】まず、ＵＳＣ２００は、利用者からの鍵生成指示にしたがい、暗号化／復号化プログラム２０４を実行して、公開鍵暗号における公開鍵と秘密鍵とを生成する（ステップ１００）。たとえば、利用者が、計算機装置２０１が備えるキーボード、マウスなどの入力装置を用いて乱数を投入すると、計算機装置２０１は、その乱数を鍵のシードとして、公開鍵暗号の公開鍵と秘密鍵とを生成し、これを計算機装置２０１が備えるハードディスクなどの記憶装置に鍵ファイルとして格納する。

【００４７】次に、ＵＳＣ２００は、公開鍵証明書申請・更新プログラム２０６の実行を開始し、ステップ１００で生成した公開鍵および秘密鍵を、ＣＡ２２０に送付する。これにより、第１の公開鍵証明書の発行を申請する（ステップ１０２）。なお、公開鍵および秘密鍵の送付方法は、ＣＡ２２０において、適当な利用者認証を利用することができるのであれば、オンラインおよびオフラインのいずれでもよい。

【００４８】ＣＡ２２０は、ＵＳＣ２００から第１の公開鍵証明書の発行申請を受け取ると、公開鍵証明書申請・更新プログラム２２６の実行を開始し、第１の公開鍵証明書を発行して、これをＵＳＣ２００に返信する。また、発行した第１の公開鍵証明書をデータベース２２２に格納する（ステップ１０４）。

【００４９】次に、ＵＳＣ２００は、ＣＡ２２０が第１の公開鍵証明書を発行したことを確認すると、当該証明書の公開鍵と対の秘密鍵を供託するか否かについて、利用者に指示を催促する。そして、受け付けた指示をＣＡ２２０に通知する（ステップ１０８）。その後、公開鍵証明書申請・更新プログラム２０６の実行を一旦終了する。

【００５０】ＣＡ２２０は、ＵＳＣ２００から受け取った通知が供託する旨を示している場合、ステップ１０２で受け取った秘密鍵と、ステップ１０４で発行した第１の公開鍵証明書（あるいは、当該証明書の識別子）とを、ＫＥＣ２３０に送付する（ステップ１１０）。その後、公開鍵証明書申請・更新プログラム２２８の実行を一旦終了する。

【００５１】ＫＥＣ２３０は、秘密鍵および第１の公開鍵証明書（あるいは、当該証明書の識別子）を受け取ると、秘密鍵供託プログラム２３５を実行し、鍵保管装置２３０ｂのいずれか一方、あるいは両方に当該秘密鍵を保管する（ステップ１１０）。この際、セキュリティをさらに向上させるために、Secret Sharingと呼ばれる暗号技術を用いて、秘密鍵を、当該秘密鍵を類推不可能なデータに分割して保管し、必要に応じて合成して、当該秘密鍵を復元するようにしてもよい。

【００５２】なお、ステップ１１０での処理は、実際には、ＣＡ２２０のバックエンドに配置されたＫＥＣ２３０が行っているが、利用者からすれば、ＣＡ２２０が秘密鍵を保管しているように見える。

【００５３】次に、ＫＥＣ２３０は、鍵供託情報テーブル参照・登録・更新プログラム２３６を実行して、鍵を供託した証拠としての付加情報（鍵供託識別子）と第１の公開鍵証明書の識別子とを関連付けて、鍵供託情報テーブル２３４に格納する。

【００５４】ここで、第三者あるいはＵＳＣ２００の利用者が、第１の公開鍵証明書の公開鍵を用いてデータ暗号を行った場合（ステップ１２０）、具体的には、セッション鍵で暗号化したデータ（ステップ１２２）に、第１の公開鍵証明書の公開鍵を用いてカプセル化したセッション鍵（ステップ１２４）を添付した場合を想定する。そして、ＵＳＣ２００の利用者がこのデータを復号する場合を考える。

【００５５】第１の公開鍵証明書の公開鍵と対の秘密鍵を紛失していない場合（ステップ１１２）、ＵＳＣ２００は、利用者からの復号指示にしたがい、暗号化／復号プログラム２０４を実行して、暗号データを復号化する。具体的には、第１の公開鍵証明書の公開鍵と対の秘密鍵を用いて、第１の公開鍵証明書の公開鍵によりカプセル化されたセッション鍵をカプセル開放し、それから、カプセル開放されたセッション鍵を用いて、暗号データを復号化する。

【００５６】一方、第１の公開鍵証明書の公開鍵と対の秘密鍵を紛失してしまった場合（ステップ１１２）、ＵＳＣ２００は、第１の公開鍵証明書の公開鍵によりカプセル化されたセッション鍵をカプセル開放することができない。

【００５７】この場合、ステップ１３０に示す公開鍵証明書再発行処理が実行されることになる。

【００５８】まず、ＵＳＣ２００は、利用者からの公開鍵証明書再発行指示にしたがい、公開鍵証明書申請・更新プログラム２０６を再度実行して、ＣＡ２２０に対し、ステップ１０４にて発行された第１の公開鍵証明書の無効化（廃止）を申請する（ステップ１３１）。

【００５９】ＣＡ２２０は、第１の公開鍵証明書の無効申請を受け取ると、公開鍵証明書申請・更新プログラム２２８を再度実行して、データベース２２２に格納された当該第１の公開鍵証明書の内容を無効状態に更新し、当該証明書の廃止をＵＳＣ２００に返信する。さらに、ＣＡ２２０は、関連部署に、ＣＲＬ（Certificate Revocation List）とよばれる通知を行う（ステップ１３２）。

【００６０】次に、ＵＳＣ２００は、利用者からの鍵生成指示にしたがい、暗号化／復号化プログラム２０４を実行して、ステップ１００と同様、公開鍵暗号における公開鍵と秘密鍵とを生成する（ステップ１３３）それから、公開鍵証明書申請・更新プログラム２０６により、ステップ１０２と同様、ステップ１３３で生成した公開鍵および秘密鍵を、ＣＡ２２０に送付する。これにより、公開鍵証明書の再発行（第２の公開鍵証明書の発行）を申請する（ステップ１３４）。なお、この再発行申請にあたり、ステップ１３２で行った第１の公開鍵証明書の廃止通知、あるいは廃止された第１の公開鍵証明書そのものを添付する必要があるかどうかは、システム運用によって異なることはいうまでもない。本実施形態では、添付しない場合について説明している。

【００６１】次に、ＣＡ２２０は、ＵＳＣ２００から公開鍵証明書の再発行申請を受け取ると、公開鍵証明書申請・更新プログラム２２６により、第２の公開鍵証明書を発行して、これをＵＳＣ２００に返信する。また、発行した第２の公開鍵証明書をデータベース２２２に格納する（ステップ１３５）。この第２の公開鍵証明書は、ステップ１３３で生成した公開鍵についての証明書である。また、ステップ１３５において、ＣＡ２２０は、第２の公開鍵証明書と結合する方法および別途送付する方法のいずれかにより、第１の公開鍵証明書の公開鍵と対の秘密鍵がＫＥＣ２３０に供託された証拠を示す付加情報および当該秘密鍵を示す鍵供託識別子を、ＵＳＣ２００に送付する。その後、フローを終了する。

【００６２】なお、図２に示すフローでは、ステップ１０２において、公開鍵および秘密鍵を一括してＣＡ２２０に提出し、第１の公開鍵証明書の発行を確認した後に、ステップ１０８において、秘密鍵の供託をするか否かを判断をするようにしている。しかしながら、ステップ１０８は、ステップ１０２の段階、すなわち、第１の公開鍵証明書の発行を申請する段階で行うようにしてもよい。あるいは、ステップ１０２では公開鍵のみを提出し、ステップ１０８において、ＵＳＣ２００が電子署名プログラム２０２を実行して、第１の公開鍵証明書の公開鍵で検証可能な署名を生成して、当該署名と共に秘密鍵を供託するようにしてもよい。

【００６３】次に、上記説明したステップ１３０（公開鍵証明書更新処理）におけるデータの流れについて説明する。

【００６４】なお、以下の図の説明では、鍵Ｋを用いてデータＸを暗号化することをＥ［Ｋ］（Ｘ）と記す。

【００６５】図３は、図２のステップ１３０（公開鍵証明書の更新手順）におけるデータの流れを概念的に示した図である。

【００６６】図３において、ＵＳＣ２００は、ＣＡ２２０に対して、第１の公開鍵証明書の無効化要求３０１を送信する。当該要求３０１には、一般にリボケーション要求と呼ばれる申請CertRevoke-reqを乱数ｒで暗号化したものと、乱数ｒをＣＡ２２０の公開鍵CApubでカプセル化したものとが含まれる。

【００６７】ＣＡ２２０は、第１の公開鍵証明書の無効化要求３０１を受信すると、ＣＡ２２０の秘密鍵CAprを用いて乱数ｒを取り出し、さらに申請CertRevoke-reqを復号する。そして、申請CertRevoke-reqの内容を審理し、ＯＫならば、ＵＳＣ２００に、無効化ＯＫの応答３０２を返す。当該応答３０２には、申請結果通知CertRevoke-resが含まれる。このとき、ＵＳＣ２００には、信頼できる公開鍵がまだ存在してない。そこで、ワンタイムな認証手続きの実現手段として、たとえば、ＵＳＣ２００から送られてきた乱数ｒを基に、申請結果通知CertRevoke-resを暗号化することが好ましい。

【００６８】次に、ＵＳＣ２００は、ＣＡ２２０に、公開鍵証明書の再発行（第２の公開鍵証明書の発行）要求３１１を送信する。当該要求３１１には、一般にリニューアル要求と呼ばれる申請CertRenewal-reqを乱数$r_2$で暗号化したものと、新しく生成した公開鍵$User_A$newPub（オプションでは秘密鍵$User_A$newpri）を乱数$r_2$で暗号化したものと、乱数$r_2$をＣＡ２２０の公開鍵CApubでカプセル化したものとが含まれる。

【００６９】ＣＡ２２０は、公開鍵証明書の再発行（第２の公開鍵証明書の発行）要求３１１を受信すると、ＣＡ２２０の秘密鍵CAprを用いて乱数$r_2$を取り出し、さらに申請CertRenewal-reqを復号する。ここで、第１の公開鍵証明書の無効化要求３０１と公開鍵証明書の再発行要求３１１とが、同じセッションで行われるのであれば、乱数$r_2$に乱数ｒと同じものを用いてもよい。

【００７０】次に、ＣＡ２２０は、復号した申請CertRenewal-reqの内容を審理し、ＯＫならば、ＫＥＣ２３０に、無効化された第１の公開鍵証明書の公開鍵と対の秘密鍵についての鍵回復有効要求３２１を送信する。当該要求３２１には、無効化された第１の公開鍵証明書の公開鍵$User_A$oldPubおよび当該公開鍵と対の秘密鍵を用いた復号を可能にするための申請KECActive-reqを、乱数$r_3$で暗号化したものと、乱数$r_3$をＫＥＣ２３０の公開鍵KECpubでカプセル化したものとが含まれる。

【００７１】ＫＥＣ２３０は、鍵回復有効要求３２１を受信すると、ＫＥＣ２３０の秘密鍵KECpriを用いて乱数$r_3$を取り出し、さらに、申請KECActive-reqを復号する。

【００７２】次に、ＫＥＣ２３０は、復号した申請KECActive-reqの内容を審理し、ＯＫならば、ＣＡ２２０に、鍵回復有効要求３２１に対する応答３２２を返信する。当該応答３２２には、申請結果通知KECActive-resが含まれる。さらに、無効化された第１の公開鍵証明書の公開鍵と対の秘密鍵を示す鍵供託識別子$KE-Id_A$が含まれる。このとき、鍵回復有効要求３２１に対する応答３２２をワンタイムに限るために、たとえば、ＣＡ２２

０から送られてきた乱数$r_3$を基に、申請結果通知KECActive-resを暗号化することが好ましい。

【００７３】また、ＫＥＣ２３０は、鍵供託情報テーブル２３４に格納された、鍵供託識別子KE－$Id_A$に対応付けられた第１の公開鍵証明書（あるいは第１の公開鍵証明書を示す識別子）の内容を、鍵回復有効状態に更新する。

【００７４】ＣＡ２２０は、ＫＥＣ２３０から鍵回復有効要求３２１に対する応答３２２を受信すると、ＵＳＣ２００に、公開鍵証明書の再発行要求３１１に対する応答３１２を返信する。当該応答３１２には、申請結果通知CertRenewal-resが含まれる。さらに、再発行した公開鍵証明書（第２の公開鍵証明書）Certficateと、鍵供託識別子KE－$Id_A$とが含まれる。このとき、第１の公開鍵証明書が廃止されたため、ＵＳＣ２００には信頼できる公開鍵が存在しないので、ワンタイムな認証手続きの実現手段として、たとえば、ＵＳＣ２００から送られてきた乱数$r_2$を基に申請結果通知CertRenewal-resを暗号化することが好ましい。

【００７５】このように、図３に示す公開鍵更新手順では、ＵＳＣ２００からの公開鍵証明書の再発行要求３１１を受け取ると、その延長として、ＫＥＣ２３０に対して鍵回復要求３２１を通知している。この点が従来の公開鍵更新手順と異なる。

【００７６】なお、図３では、第１の公開鍵証明書の無効化と第２の公開鍵証明書の発行（公開鍵証明書の再発行）との両方を確認した後に鍵回復を有効にするため、公開鍵証明書の再発行要求３１１に応答して、鍵回復有効要求３２１を起動するようにしている。しかしながら、公開鍵証明書の再発行要求３１１ではなく、第１の公開鍵証明書の無効化要求３０１に応答して、鍵回復有効要求３２１を起動するようにしてもよい。

【００７７】以上、第１の公開鍵証明書を発行してから、当該証明書を廃止して第２の公開鍵証明書を再発行するまでの処理手順について説明した。

【００７８】次に、利用者が、再発行を受けた第２の公開鍵証明書を用いて認証を受けることで、廃止した第１の公開鍵証明書の公開鍵でカプセル化されたセッション鍵をカプセル開放してもらうまでの処理手順について、図４を参照して説明する。

【００７９】図４は、本発明の第１実施形態において、利用者を認証してから、廃止した第１の公開鍵証明書の公開鍵でカプセル化されたセッション鍵をカプセル開放するまでの処理手順を説明するためのフロー図である。

【００８０】まず、ＵＳＣ２００は、利用者から廃止した第１の公開鍵証明書の公開鍵でカプセル化されたセッション鍵の開放要求を受け取ると、暗号データ鍵回復・再暗号化プログラム２０８を実行して、対象となるエンベロープデータ（廃止された第１の公開鍵証明書の公開鍵でカプセル化されたセッション鍵と、当該セッション鍵で暗号化されたデータとを含む）、および、ＣＡ２２０から受け取った、廃止された第１の公開鍵証明書の公開鍵と対の秘密鍵を示す鍵供託識別子KE－$Id_A$からなる鍵回復要求の申請を作成する。次に、電子署名プログラム２０２を実行して、ＣＡ２２０が再発行した公開鍵証明書（第２の公開鍵証明書）の公開鍵と対の秘密鍵を用いて、前記申請に署名してＤＲＣ２４０に送信する（ステップ１４０）。

【００８１】ＤＲＣ２４０は、鍵回復要求の申請を受け取ると、回復者認証プログラム２４２を実行し、当該申請書に付された公開鍵証明書（ＣＡ２２０が再発行した第２の公開鍵証明書）に記された公開鍵を用いて、当該申請書の署名を検証する（ステップ１４２）。その結果、署名が当該申請書に付された公開鍵証明書の公開鍵と対の秘密鍵によるものでないことが判明した場合は、エラー処理を行う（ステップ１４８）。

【００８２】一方、署名が当該申請書に付された公開鍵証明書の公開鍵と対の秘密鍵によるものであることが判明した場合は、当該申請書の送信者が、鍵回復権限をもつ利用者であるか否かを必要に応じて検証する（ステップ１４４）。この鍵回復権限をもつ利用者であるか否かの検証は、たとえば、鍵供託識別子を用いて行う。本実施形態では、図２のステップ１３５において、ＣＡ２２０がＵＳＣ２００に送付された鍵供託識別子KE－$Id_A$を、ステップ１４０において、ＵＳＣ２００からＤＲＣ２４０へ送信している。そこで、ＤＲＣ２４０は、ステップ１４２において、署名が申請書に付された公開鍵証明書の公開鍵と対の秘密鍵によるものであることが判明した場合、当該申請書に付された鍵供託識別子KE－$Id_A$をＫＥＣ２３０に送信する。これを受けて、ＫＥＣ２３０は、鍵供託情報テーブル参照・登録・更新プログラム２３６を実行し、鍵供託情報テーブル２３４を参照して、受け取った鍵供託識別子KE－$Id_A$に対応付けられた公開鍵証明書（第１の公開鍵証明書）が鍵回復をしてもよい（鍵回復有効）状態にあるか否かを判定する。そして、その判定結果を回復者認証プログラム２４２の戻り値に含めて、利用者認証を行う。

【００８３】ステップ１４４において、回復権限をもつ利用者として認証された場合、すなわち、利用者が提供した鍵供託識別子KE－$Id_A$を用いた鍵供託情報テーブル２３４の参照より、ＫＥＣ２３０に鍵が保管され且つ鍵回復をしてもよい状態であると判定された場合には、ステップ１５０に示すセッション鍵回復処理が行われる（ステップ１４６）。一方、回復権限をもつ利用者として認証されなかった場合は、ステップ１４８に示すエラー処理が行われる（ステップ１４６）。

【００８４】ステップ１５０のセッション鍵回復処理において、ＤＲＣ２４０は、暗号データ鍵回復・暗号化プログラム２４３の実行を開始し、ＫＥＣ２３０に対して、鍵供託識別子KE－$Id_A$により特定される鍵（第１の

公開鍵証明書の公開鍵と対の秘密鍵）の検索を依頼する。これを受けて、ＫＥＣ２３０は、秘密鍵検索プログラム２３７を実行して；鍵供託識別子KE－$Id_A$により特定される鍵を検索する（ステップ１５２）。本実施形態では、供託鍵識別子がユニークであることを前提としているので、供託鍵識別子自体、あるいは当該供託鍵識別子から導出される識別子（保管鍵識別子）に基づいて検索することができる。

【００８５】次に、ＤＲＣ２４０は、ステップ１５２で検索された、第１の公開鍵証明書の公開鍵と対の秘密鍵を用いて、ＵＳＣ２００から送られてきた鍵回復要求の申請書に付されたセッション鍵（第１の公開鍵証明書の公開鍵でカプセル化された鍵）を復号する（ステップ１５４）。

【００８６】次に、ＤＲＣ２４０は、鍵回復要求の申請書に付されたセッション鍵のカプセル開放のみならず、当該申請書に付された暗号データの復号をも行うか否か判断する（ステップ１６０）。この判断は、たとえば、鍵回復要求の申請書にカプセル化されたセッション鍵および当該セッション鍵で暗号化されたデータの両方が付されている場合は、暗号データの復号を行うものと判断し、カプセル化されたセッション鍵のみが付されている場合は、暗号データの復号を行わないものと判断するようにしてもよい。

【００８７】鍵回復要求の申請書に付された暗号データの復号を行うと判断した場合は、ステップ１７０に示す暗号データ復号・再暗号化処理が行われる。一方、行わないと判断した場合は、フローを終了する。

【００８８】ステップ１７０において、ＤＲＣ２４０は、暗号データ鍵回復・暗号化プログラム２４３により、ステップ１５４で復号されたセッション鍵を用いて、鍵回復要求の申請書に付された暗号データを復号する（ステップ１７２）。次に、再暗号化用のセッション鍵を生成し（ステップ１７４）、ステップ１７２で復号化されたデータ（平文）を、前記再暗号化用のセッション鍵を用いて暗号化する（ステップ１７６）。さらに、ＵＳＣ２００の利用者に対して、ＣＡ２２０が再発行した第２の公開鍵証明書の公開鍵を用いて、前記再暗号化用のセッション鍵をカプセル化し、これをステップ１７６で暗号化したデータに添付する。これにより、エンベロープデータを生成する（ステップ１７８）。そして、生成したエンベロープデータをＵＳＣ２００に送付して、フローを終了する。

【００８９】なお、図４に示すフローでは、エンベロープデータを対象にしたものについて説明したが、セッション鍵と暗号データとを別途に送る暗号方式のスキームでも、同様に処理できることはいうまでもない。

【００９０】次に、上記説明した図４のフローのステップ１４４～ステップ１７０（鍵回復手順）におけるデータの流れについて説明する。

【００９１】図５は、図４に示すフローのステップ１４４～ステップ１７０（鍵回復手順）におけるデータの流れを概念的に示した図である。

【００９２】まず、ＵＳＣ２００は、ＤＲＣ２４０に対して、廃止した第１の公開鍵証明書の公開鍵と対の秘密鍵を用いた鍵回復が可能な状態にあるか否かの状態問い合わせである鍵回復可能確認要求要求４０１を送信する。当該要求４０１には、廃止した第１の公開鍵証明書の公開鍵と対の秘密鍵を用いた鍵回復が可能な状態にあるか否かの状態問い合わせ申請?CanKRS-reqと供託鍵識別子KE-$Id_A$とを乱数ｒで暗号化したものと、乱数ｒをＤＲＣ２４０の公開鍵DRCpubでカプセル化したものとが含まれる。

【００９３】ＤＲＣ２４０は、鍵回復可能確認要求要求４０１を受信すると、ＤＲＣの秘密鍵DRCpriを用いて乱数ｒを取り出し、さらに申請?Can KRS-reqおよび供託鍵識別子KE-$Id_A$を復号する。そして、ＫＥＣ２３０に、供託鍵識別子KE-$Id_A$が示す供託鍵（第１の公開鍵証明書の公開鍵と対の秘密鍵）を用いた鍵回復が、有効な状態にあるか否かの問い合わせである供託鍵ＩＤ有効確認要求４２１を送信する。当該要求４２１には、供託鍵識別子KE-$Id_A$を用いた鍵回復が有効であるか否かを問い合わせるための申請KECStatus-reqと供託鍵識別子KE-$Id_A$とを乱数$r_2$で暗号化したものと、乱数$r_2$をＫＥＣ２３０の公開鍵KECpubでカプセル化したものとが含まれる。

【００９４】ＫＥＣ２３０は、供託鍵ＩＤ有効確認要求４２１を受信すると、ＫＥＣの秘密鍵KECpriを用いて乱数$r_2$を取り出し、さらに申請KECStatus-reqおよび供託鍵識別子KE-$Id_A$を復号する。そして、鍵供託情報テーブル２３４を参照して、復号した鍵供託識別子KE－$Id_A$に対応付けられた公開鍵証明書（第１の公開鍵証明書）が鍵回復をしてもよい（鍵回復有効）状態にあるか否かを判定する。そして、ＤＲＣ２４０に、供託鍵ＩＤ有効確認要求４２１に対する応答４２２を返信する。この応答には、鍵供託識別子KE－$Id_A$が示す秘密鍵（第１の公開鍵証明書の公開鍵と対の秘密鍵）による鍵回復についての有効／無効の判定結果通知KECStatus-resが含まれる。このとき、ワンタイムな認証手続きの実現手段として、たとえば、ＤＲＣから送られてきた乱数$r_2$を基に、判定結果通知KECStatus-resを暗号化することが好ましい。

【００９５】ＤＲＣ２４０は、ＫＥＣ２３０から応答４２２を受信すると、ＵＳＣ２００に、鍵回復可能確認要求要求４０１に対する応答４０２を返信する。当該応答４０２には、ＫＥＣ２３０からの応答４２２に含まれる判定結果通知KECStatus-resに対応した申請結果通知?CanKRS-resと、鍵供託識別子KE－$Id_A$と、乱数$r_3$とが含まれる。このとき、ワンタイムな認証手続きの実現手段として、たとえば、ＵＳＣ２００から送られてきた乱数ｒを基に、申請結果通知?CanKRS-res、鍵供託識別子KE－I

$d_A$、および乱数$r_3$を暗号化することが好ましい。あるいは、ＵＳＣ２００の利用者に対して発行された第２の公開鍵証明書の公開鍵$User_A$newpubを用いて暗号してもよい。

【００９６】次に、ＵＳＣ２００は、ＤＲＣ２４０からの応答４０２に含まれる申請結果通知?CanKRS-resが、鍵供託識別子KE－$Id_A$で特定される鍵を用いた鍵回復が有効であることを示している場合、ＤＲＣ２４０に、鍵供託識別子KE－$Id_A$で特定される鍵を用いた鍵回復要求４１１を送信する。当該要求４１１には、暗号データCiphertext(平文をセッション鍵SKで暗号化したものと、セッション鍵SKを供託識別子KE-$Id_A$が示す鍵（第１の公開鍵証明書の公開鍵と対の秘密鍵$User_A$oldpri)でカプセル化したものとを含む)と、鍵回復要求を示す申請KeyRecovery-reqおよび供託鍵識別子KE-$Id_A$を、応答４０２に含まれる乱数$r_3$で暗号化したものと、乱数$r_3$をＤＲＣ２４０の公開鍵DRCpubでカプセル化したものとが含まれる。

【００９７】ＤＲＣ２４０は、ＵＳＣ２００から鍵回復要求４１１を受信すると、ＤＲＣ２４０の秘密鍵DRCpriを用いて乱数$r_3$を取り出し、さらに申請KeyRecovery-reqおよび供託鍵識別子KE-$Id_A$を復号する。ここで、上記説明した鍵回復可能確認要求要求４０１と鍵回復要求４１１とが、同じセッションで行われるのであれば、乱数$r_3$は、鍵回復可能確認要求要求４０１で用いた乱数ｒと同じものでもよい。

【００９８】次に、ＤＲＣ２４０は、ＫＥＣ２３０から受け取った応答４２２に含まれる、供託鍵識別子KE-$Id_A$により特定される鍵を用いた鍵復号についての有効／無効の判定結果通知KECStatus-resを調べ、該通知の内容が有効であれば、ＫＥＣ２３０に、セッション鍵回復要求４３１を送信する。当該要求４３１には、セッション鍵SKを第１の公開鍵証明書の公開鍵$User_A$oldPubでカプセル化したもの、復号化することを許可することを示す申請KECReocvery-req、および、供託鍵識別子KE-$Id_A$を乱数$r_4$で暗号化したものと、乱数$r_4$をＫＥＣ２３０の公開鍵KECpubでカプセル化したものとが含まれる。

【００９９】ＫＥＣ２３０は、ＤＲＣ２４０からセッション鍵回復要求４３１を受信すると、ＫＥＣ２３０の秘密鍵KECpriを用いて乱数$r_4$を取り出し、さらに申請KECReocvery-req、供託鍵識別子KE-$Id_A$、および公開鍵$User_A$oldPubでカプセル化したセッション鍵SKを復号する。次に、供託鍵識別子KE-$Id_A$により特定される鍵（第１の公開鍵証明書の公開鍵と対の秘密鍵$User_A$oldpri)を検索し、この鍵を用いて、セッション鍵SKをカプセル開放する。

【０１００】それから、ＫＥＣ２３０は、ＤＲＣ２４０に、セッション鍵回復要求４３１に対する応答４３２を返信する。当該応答４３２には、鍵回復が行われたか否かを示す申請結果通知KECReocvery-resと、カプセル開放されたセッション鍵SKとが含まれる。このとき、セッション鍵回復要求４３１に対する応答４３２をワンタイムに限るために、たとえば、ＤＲＣ２４０から送られてきた乱数$r_4$を基に、申請結果通知KECActive-res、およびセッション鍵SKを暗号化することが好ましい。

【０１０１】ＤＲＣ２４０は、ＫＥＣ２３０から応答４３２を受信すると、当該応答４３２に含まれるセッション鍵SKを用いて、ＵＳＣ２００からの鍵回復要求４１１に含まれる暗号データCiphertextを平文Plaintextに復号する。それから、ＵＳＣ２００に、鍵回復要求４１１に対する応答４１２を返信する。当該応答４１２には、ＫＥＣ２３０から受け取った応答４３２に含まれる申請結果通知KeyRecovery-resと、平文Plaintextを乱数$r_5$で暗号化したものと、乱数$r_5$をＵＳＣ２００の利用者に再発行された第２の公開鍵証明書の公開鍵$User_A$newPubでカプセル化したものとが含まれる。このときに，ワンタイムな認証手続きの実現手段として、たとえば、ＵＳＣ２００から送られてきた乱数$r_3$を基に、申請結果通知KeyRecovery-resを暗号化することが好ましい。

【０１０２】このように、図５に示す鍵回復手順では、ＤＲＣ２４０は、ＵＳＣ２００からの鍵回復可能確認要求４０１を受け取ると、当該要求４０１に付された供託鍵識別子を用いて、当該識別子により特定される鍵（第１の公開鍵証明書の公開鍵と対の秘密鍵）を用いた鍵回復が有効であるか否かの判定を行うようにしている。そして、有効である場合には、当該鍵を用いてセッション鍵をカプセル開放し、当該セッション鍵で暗号化されたデータを復号化するとともに、当該復号化されたデータを新たなセッション鍵で再暗号化し、さらに、第１の公開鍵証明書を廃止した際に発行された第２の公開鍵証明書の公開鍵を用いて、当該新たなセッション鍵を再カプセル化している。この点が従来の鍵回復手順と異なる。

【０１０３】以上、利用者が、再発行を受けた第２の公開鍵証明書を用いて認証を受けることで、廃止した第１の公開鍵証明書の公開鍵でカプセル化されたセッション鍵をカプセル開放してもらうまでの処理手順について説明した。

【０１０４】上記の実施形態では、ＣＡ２２０は、第１の公開鍵証明書（データ暗復号用）を発行した利用者が当該証明書の公開鍵と対の秘密鍵を紛失した場合、当該利用者から要求があった場合にのみ、当該証明書を廃止して、前記利用者に第２の公開鍵証明書（データ暗復号兼認証用）を発行するようにしている。

【０１０５】そして、ＤＲＣ２４０は、前記第２の公開鍵証明書を用いて利用者を認証することにより、前記第２の公開鍵証明書を発行を受けた利用者にのみ、当該利用者が第１の公開鍵証明書の発行の際にＫＥＣ２３０に供託した公開鍵と対の秘密鍵を用いた鍵回復を認めるようにしている。

【０１０６】このように、本実施形態では、鍵回復権限

を有する利用者を当該利用者に発行した第２の公開鍵証明書を用いて認証するので、鍵回復の利用者を登録する必要がない。すなわち、公開鍵証明書の利用者の認証・登録と鍵回復の利用者の認証・登録とを１元管理することができる。また、ＣＡ２２０による公開鍵証明書の発行を受けた利用者に対してのみ、当該公開鍵証明書の公開鍵と対の秘密鍵を紛失した場合に、当該秘密鍵を用いた鍵回復を認めることができる。

【０１０７】また、本実施形態では、ＤＲＣ２４０は、鍵回復権限を有する利用者の要求により、回復したセッション鍵を用いて暗号データを復号した後、この復号したデータを新たなセッション鍵で再暗号化し、さらに、この新たなセッション鍵を、ＣＡ２２０が当該利用者に再発行した第２の公開鍵証明書の公開鍵を用いてカプセル化して、当該利用者のＵＳＣ２００に送信するようにしている。

【０１０８】このようにすることで、回復したデータを利用者に通知する際のセキュリティを向上させることができる。

【０１０９】以上、本発明の第１実施形態について説明した。

【０１１０】次に、本発明の第２実施形態について説明する。

【０１１１】上記の第１実施形態では、利用者を当該利用者に発行した第２の公開鍵証明書を用いて認証し、認証された場合に鍵回復を認めている。これに対し、本実施形態では、鍵回復の有効期限を設定しておき、認証された場合でも、有効期限が切れている場合は鍵回復を認めないようにしている。

【０１１２】本実施形態は、上記の第１実施形態に対し、図２に示すステップ１３５での処理と、図４に示すステップ１５２での処理が異なる。その他については、上記の第１実施形態と同じであるので、図２に示すステップ１３５および図４に示すステップ１５２に対応する処理についてのみ説明し、その他の説明は省略する。

【０１１３】図６は、本発明の第２実施形態における、図２のステップ１３５に対応するステップ１３５ａでの処理を説明するためのフロー図である。

【０１１４】まず、ＣＡ２２０は、ＵＳＣ２００から公開鍵証明書の再発行申請を受け取ると、公開鍵証明書申請・更新プログラム２２６の実行を開始し、管理者が、鍵回復の期間について、あらかじめ定義したシステム設定を読み込む（ステップ５００）。このシステム設定とは、ファイルの形で与えられ、有効期限データ、有効期限データの送付方法（公開鍵証明書の拡張をする／しない、その他の方法）、鍵回復を認めない利用者のリスト等を記述したものである。このシステム設定ファイルは、計算機装置２２１の記憶装置に、あらかじめ格納されているものとする。

【０１１５】次に、ＣＡ２２０は、ステップ５００で読み込んだシステム設定にしたがい、有効期限を公開鍵証明書に添付するか否かを判定する（ステップ５０２）。添付しない場合は、別途送付を選択し（ステップ５１０）。有効期限を記憶媒体に記憶してオフラインでＵＳＣ２００の利用者の送付するか、あるいは、公開鍵証明書の送信とは別に、オンラインでＵＳＣ２００に送信する。この際、ＣＡ２２０の秘密鍵を用いて署名をしておくことが好ましい。

【０１１６】一方、公開鍵証明書に添付する場合は、さらに、公開鍵証明書の形式を拡張するか否かの判定を行なう（ステップ５０４）。

【０１１７】公開鍵証明書の形式を拡張する場合は、公開鍵証明書を作成するとともに、当該公開鍵証明書に、公開鍵証明書の拡張形式として鍵回復の有効期限に関する情報を追加する（ステップ５０６）。たとえば、公開鍵証明書x.５０９Version３の拡張形式extensionsとして、供託鍵識別子KeyEscrowIdentifierおよび鍵回復の有効期限KeyRecvoeryUsagePeriodの情報を添付する。

【０１１８】公開鍵証明書の形式を拡張しない場合は、公開鍵証明書を作成するとともに、当該公開鍵証明書の有効期限Validityを鍵回復の有効期限KeyRecvoeryUsagePeriodの代わりとする（ステップ５０７）。

【０１１９】次に、ＣＡ２２０は、作成した公開鍵証明書を第２の公開鍵証明書として再発行して、これをＵＳＣ２００に返信する。また、発行した第２の公開鍵証明書をデータベース２２２に格納する（ステップ５０８）。また、ＣＡ２２０は、第２の公開鍵証明書と結合する方法および別途送付する方法のいずれかにより、第１の公開鍵証明書の公開鍵と対の秘密鍵がＫＥＣ２３０に供託された証拠を示す付加情報および当該秘密鍵を示す鍵供託識別子を、ＵＳＣ２００に送付する。

【０１２０】上記のフローにより、公開鍵証明書の再発行に際し、鍵回復の有効期限が設定されることになる。

【０１２１】なお、上記のステップ５０７において、公開鍵証明書の有効期限を、そのまま鍵回復の有効期限とするのではなく、評価式、たとえば、公開鍵証明書に記された有効期限の１／１２の期間が経過するまで（公開鍵証明書の有効期限が１年であれば最初の３０日が経過するまで）を鍵回復の有効期限とする評価式を、ＣＡ２２０とＤＲＣ２４０との間で取り交わしておくようにしてもよい。

【０１２２】また、上記のステップ５１０において、ＣＡ２２０は、自らの署名が付いた鍵回復の有効期限の情報を、ＵＳＣ２００に送付している。したがって、ＵＳＣ２００の利用者は、ＣＡ２２０の署名が付いた有効期限の情報に、さらに自らの署名を付けて、ＤＲＣ２４０に通知することになる。しかしながら、ステップ５１０において、ＣＡ２２０の署名が付いた鍵回復の有効期限の情報を、ＣＡ２２０からＤＲＣ２４０に直接送付するようにしてもよい。

【０１２３】図７は、本発明の第２実施形態における、図４のステップ１５２に対応するステップ１５２ａでの処理を説明するためのフロー図である。

【０１２４】まず、ＤＲＣ２４０は、暗号データ鍵回復・再暗号化プログラム２４３の実行を開始し、管理者が、鍵回復の期間について、あらかじめ定義したシステム設定を読み込む（ステップ５５０）。このシステム設定とは、ステップ５００でのシステム設定と同じものであり、ファイル形式で、計算機装置２４１の記憶装置に、あらかじめ格納されているものとする。

【０１２５】次に、ＤＲＣ２４０は、ステップ５５０で読み込んだシステム設定にしたがい、鍵回復の有効期限が公開鍵証明書に添付されている否かを判定する（ステップ５５２）。添付されていない場合は、別途送付を選択し（ステップ５６０）。ＵＳＣ２００の利用者から有効期限を記憶した記憶媒体がオフラインで送付されてくるか、あるいは、公開鍵証明書とは別に、オンラインで送信されてくるのを待つ。この際、ＵＳＣ２２０の利用者は、たとえば再発行された第２の公開鍵証明書の公開鍵と対の秘密鍵を用いて署名をしておくことが好ましい。

【０１２６】一方、鍵回復の有効期限が公開鍵証明書に添付されている場合は、さらに、公開鍵証明書の形式が拡張されているか否かの判定を行なう（ステップ５５４）。

【０１２７】公開鍵証明書の形式が拡張されている場合は、第２の公開鍵証明書に拡張形式として追加された鍵回復の有効期限に関する情報を読み込む（ステップ５５６）。たとえば、公開鍵証明書x.５０９Version３の拡張形式extensionsとして添付された、供託鍵識別子KeyEscrowIdentifierおよび鍵回復の有効期限KeyRecvoeryUsagePeriodの情報を読み込む。

【０１２８】公開鍵証明書の形式が拡張されていない場合は、第２の公開鍵証明書の有効期限Validityを鍵回復の有効期限KeyRecvoeryUsagePeriod、あるいは鍵回復の有効期限を特定するためのパラメータとして読み込む（ステップ５５７）。

【０１２９】次に、ＤＲＣ２４０は、上記のステップにより得られた鍵回復の有効期限を経過しているか否かを判定し、経過していない場合にのみ、ＫＥＣ２３０に対して、鍵供託識別子$KE-Id_A$により特定される鍵（第１の公開鍵証明書の公開鍵と対の秘密鍵）の検索を依頼する。これを受けて、ＫＥＣ２３０は、秘密鍵検索プログラム２３７を実行して、鍵供託識別子$KE-Id_A$により特定される鍵を検索する（ステップ１５２）。上述したように、供託鍵識別子はがユニークであることを前提としているので、供託鍵識別子自体、あるいは当該供託鍵識別子から導出される識別子（保管鍵識別子）に基づいて検索することができる。

【０１３０】上記のフローにより、鍵回復の要求に際し、鍵回復の有効期限内にある場合にのみ、鍵回復が行われることになる。

【０１３１】以上、本発明の第２実施形態について説明した。

【０１３２】次に、本発明の第３実施形態について説明する。

【０１３３】上記の第２実施形態では、鍵回復の有効期限の情報をＣＡ２２０が再発行した第２の公開鍵証明書に添付、あるいは、鍵回復の有効期限自体を別途送付することで、鍵回復の有効期限を設定するようにしている。これに対し、本実施形態では、供託鍵識別子を用いて有効期限を設定するようにしている。

【０１３４】本実施形態は、上記の第１実施形態に対し、図４に示すステップ１５０での処理が異なる。その他については、上記の第１実施形態と同じであるので、図４のステップ１５０に対応する処理についてのみ説明し、その他の説明は省略する。

【０１３５】図８は、本発明の第３実施形態における、図４のステップ１５０に対応するステップ１５０ａでの処理を説明するためのフロー図である。

【０１３６】図８に示すステップ１５０ａが図４のステップ１５０と異なる点は、ステップ１５２の前処理としてステップ６００を設けた点である。そこで、以下では、ステップ６００での処理についてのみ説明する。

【０１３７】まず、ＤＲＣ２４０は、暗号データ鍵回復・暗号化プログラム２４３の実行を開始し、鍵供託識別子$KE-Id_A$について、鍵回復の有効期限が無制限に設定されているか否かを判定する（ステップ６０１）。判定方法としては、図２に示すステップ１３５での処理において、所定の基準にしたがい鍵回復の有効期限を無制限／制限ありに設定して、この情報を供託鍵識別子に付加してＵＳＣ２００に送付しておき、ＵＳＣ２００から受け取った供託鍵識別子から当該鍵回復の有効期限情報を取り出して判断するか、あるいは、計算機装置２２１が備える記憶装置に、ＵＳＣ２００に送付した供託鍵識別子と、所定の基準にしたがい無制限／制限ありに設定された鍵回復の有効期限情報とを対応付けて記憶しておき、ＵＳＣ２００から受け取った供託鍵識別子に対応する鍵回復の有効期限情報を当該記憶装置から取り出して判断する方法などが考えられる。

【０１３８】ステップ６０１において、鍵回復の有効期限情報が無制限である場合には、ステップ１５２に移行する。一方、無制限でない場合は、現在の時刻（システムの運用規則によっては、利用者が鍵回復を要求した時刻でもよい）Tcurrentと、供託停止後、継続して供託鍵を保管する義務がある期間として予め定められた期間Tlimitとを取得する（ステップ６１２）。次に、供託鍵識別子の供託開始時刻ID-t１および停止時刻ID-t２を取得する（ステップ６１４）。ここで、供託鍵識別子の供託開始時刻ID-t１とは、ＣＡ２２０が第１の公開鍵証明書

を発行した時刻、あるいは、ＫＥＣ２３０が第１の公開鍵証明書の公開鍵と対の秘密鍵についての供託識別子を発行した時刻を意味し、供託停止時刻ID-t２とは、ＣＡ２２０が第１の公開鍵証明書を廃止した時刻、あるいは、ＫＥＣ２３０が、ＣＡ２２０による第２の公開鍵証明書の再発行に際して、１の公開鍵証明書の公開鍵と対の秘密鍵を用いた鍵回復が有効となるように、供託鍵情報テーブル２３４を更新した時刻を意味する。

【０１３９】次に、ＤＲＣ２４０は、鍵回復の対象となる暗号データが暗号化された時刻ｔを取得する（ステップ６１６）。たとえば、暗号データがファイル形式で格納されているのであれば、そのファイルの作成日時を入力する。

【０１４０】次に、ＤＲＣ２４０は、ステップ６１６で取得した時刻ｔが、供託開始時刻ID-ｔ１から供託停止時刻ID-ｔ２に供託停止後の保管義務がある期間Tlimitを加算した期間までの範囲に含まれているか否かを判定する（ステップ６１８）。

【０１４１】含まれいない場合は、鍵回復不可としてエラー処理を行う（ステップ６３０）。一方、含まれている場合は、ステップ６１２で取得した時刻Tcurrentが、供託停止時刻ID-ｔ２から供託停止時刻ID-ｔ２に供託停止後の保管義務がある期間Tlimitを加算した期間までの範囲に含まれているか否かを判定する（ステップ６１９）。含まれていない場合は、鍵回復不可としてエラー処理を行う（ステップ６３０）。一方、含まれている場合は、鍵回復を許可するものとし（ステップ６２０）、ステップ１５２に移行する。

【０１４２】上記のフローにより、鍵回復の要求に際し、鍵回復の有効期限内にある場合にのみ、鍵回復が行われることになる。

【０１４３】なお、上記のフローでは、鍵回復の有効期限を一律に設定しているが、当該期限は個々のシステム運用に応じて変形するようにしてもよいことはいうまでもない。たとえば、以下のように変形してもよい。

【０１４４】（ａ）オンラインでの回復は一律に設定される有効期限内に限定とし、オフラインでの回復は、専門の職員による面接で判断する。

【０１４５】（ｂ）特定の権限(たとえば、公開鍵証明書に任意のクラス以上の権限を有することが示されている場合)をもつ利用者に対しては、鍵回復の有効期限（すなわち、供託停止後の保管義務がある期間Tlimit）を長く設定しておく。

【０１４６】（ｃ）特定の権限をもたない利用者に対しては、オフラインでの回復を原則とし、本処理の適応範囲外とする。

【０１４７】（ｄ）暗号データの種類に応じて鍵回復の有効期限（すなわち、供託停止後の保管義務がある期間Tlimit）の設定を変える。

【０１４８】本実施形態によれば、秘密鍵の供託開始／停止は、利用者の責任で処理することができる。また、供託停止後、供託鍵を保管する期間の設定は、ＤＲＣ２４０の管理者の責任で定めることができる。また、利用者がＫＥＣ２３０に供託した鍵を紛失した場合にのみ、当該利用者に対して、当該鍵を用いた鍵回復を提供することができる。

【０１４９】以上、本発明の第３実施形態について説明した。

【０１５０】次に、本発明の第４実施形態について説明する。

【０１５１】公開鍵証明書には、通常、有効期限が設定されており、当該期限が過ぎると、当該公開鍵証明書は廃止されて、第２世代の公開鍵証明書が発行される。すなわち、上記の実施形態でいえば、データ暗復号用の第１の公開鍵証明書が多世代にわたり存在することが考えられる。

【０１５２】このような状況下で、利用者が多世代にわたり複数発行された第１の公開鍵証明書の公開鍵と対の秘密鍵各々をＫＥＣ２３０に供託する場合、供託する秘密鍵各々に対して別個の供託鍵識別子を付して管理するよりも、同じ供託鍵識別子を用いて管理することが好ましい。

【０１５３】供託鍵識別子は、鍵回復の権限を有することを保証するための機密データであり、ＩＣカードなどに格納して保管することが望まれる。しかしながら、ＩＣカードなどの媒体の記憶容量では、供託鍵識別子および公開鍵証明書(たとえば、５１２BYTE)を複数格納することが困難な場合（たとえば１K BYTE）が多い。このため、最初に発行された第１の公開鍵証明書の公開鍵と対の秘密鍵を供託した際に発行された供託鍵識別子をもとに、２回目以降に発行された第１の公開鍵証明書の公開鍵と対の秘密鍵の供託を管理することが望まれる。

【０１５４】また、利用者にとっては、相手から、多世代にわたり複数発行された第１の公開鍵証明書のうちの１つの公開鍵を用いて暗号化されたデータを受け取った場合、対応する秘密鍵が分からなくなってしまいデータを復号できなってしまう場合も考えられる。

【０１５５】本実施形態では、このような場合を想定したものであり、多世代にわたり複数発行された第１の公開鍵証明書の公開鍵と対の秘密鍵各々について、別個の供託鍵識別子を付すことなく、これらの秘密鍵の供託を管理することができるようにしている。また、利用者が多世代にわたり複数発行された第１の公開鍵証明書の公開鍵と対の秘密鍵をＫＥＣ２３０に供託している場合に、当該利用者が鍵回復を行うための秘密鍵が分からない場合でも鍵回復できるようにしている。

【０１５６】本実施形態は、上記の第１実施形態に対し、図２に示すステップ１１０での処理において、多世代にわたり複数発行された第１の公開鍵証明書（データ暗復号用）の公開鍵と対の秘密鍵各々を供託する場合

に、これら秘密鍵に同じ供託鍵識別子を付加する点、および、図４に示すステップ１５２での処理が異なる。その他については、上記の第１実施形態と同じであるので、図４のステップ１５２に対応する処理についてのみ説明し、その他の説明は省略する。

【０１５７】図９は、本発明の第４実施形態における、図４のステップ１５２に対応するステップ１５２ｂでの処理を説明するためのフロー図である。

【０１５８】まず、ＫＥＣ２３０は、秘密鍵検索プログラム２３７を実行して、ＵＳＣ２００から受け取った供託鍵識別子に複数の版があるか否かを判定する（ステップ７１０）。判定方法としては、図２に示すステップ１１０での処理において、多世代にわたり複数発行された第１の公開鍵証明書の公開鍵と対の秘密鍵各々に同じ供託鍵識別子を付加する際に、当該供託鍵識別子に版（世代）情報を含めておき、ＵＳＣ２００から受け取った供託鍵識別子に版情報が含まれるか否かで判断する方法や、ＵＳＣ２００から受け取った供託鍵識別子と同じもの（ただし、版情報が違う）がＫＥＣ２３０の供託鍵情報テーブル２３４に複数格納されているか否かで判断する方法などが考えられる。

【０１５９】ステップ７１０において、版情報がない（たとえば、版番号が０に設定されている）場合は、ステップ７００へ移行して、図４のステップ１５２と同じ処理を行い、ＵＳＣ２００から受け取った供託鍵識別子により特定される秘密鍵を検索する。具体的には、ＵＳＣ２００から受け取った供託鍵識別子を入力変数とする関数ｆ（）を計算し、秘密鍵の保管位置（たとえば、計算機装置２３２が備えるデータベースのレコード番号）を決定し、決定された保管位置から秘密鍵を取得する（ステップ７００）。ここで、関数ｆ()として、たとえば、一方向性関数が用いられる。なお、ハッシュ表探索として、保管位置が登録・参照できることが好ましい。

【０１６０】一方、ステップ７１０において、版情報がある（たとえば、版番号が１以上に設定されている）場合は、ステップ７１２に移行して、鍵回復の対象となる暗号データが暗号化された時刻ｔを取得する。たとえば、暗号データがファイル形式で格納されているのであれば、そのファイルの作成日時を入力する。

【０１６１】次に、ＵＳＣ２００から受け取った供託鍵識別子について、１からｎ番目までの版うち、どの版の秘密鍵を検索すべきかシーケンシャルに解析する。

【０１６２】まず、ＫＥＣ２３０は、鍵供託情報テーブル参照・登録・更新プログラム２３６を実行して鍵供託情報テーブル２３４を参照し、ＵＳＣ２００から受け取った供託鍵識別子に対して、各版番号ｉ＝１～ｎ毎に、鍵回復が許されるかどうかを判定する（ステップ７１４）。鍵回復が有効であれば、その版番号ｉが付された供託鍵識別子の登録時刻$T_i$と、つぎの版番号ｉ＝ｉ＋１が付された供託鍵識別子の登録時刻$T_{i+1}$とを取得する（ステップ７１６）。一方、無効であれば、つぎの版番号ｉ＝ｉ＋１についての情報を取得するため、ステップ７２２に移行する。

【０１６３】ステップ７１６での処理終了後、ＫＥＣ２３０は、ステップ７１６で取得した、版番号ｉが付された供託鍵識別子の登録時刻$T_i$と、版番号ｉ＋１が付された供託鍵識別子の登録時刻$T_{i+1}$との間に、ステップ７１２で取得した時刻ｔが含まれるか否かを判断する（ステップ７１８）。含まれていれば、版番号ｉの供託鍵情報識別子により特定される秘密鍵を用いた鍵回復が可能であると判定して、ステップ７２０に移行する。

【０１６４】一方、含まれていない場合は、版番号ｉの供託鍵情報識別子により特定される秘密鍵を用いた鍵回復が不可能であると判定し、つぎの版番号ｉ＝ｉ＋１についての情報を取得するため、ステップ７２２に移行する。

【０１６５】ステップ７２０では、ＫＥＣ２３０は、版番号ｉの供託鍵識別子と当該識別子の登録時刻Ｔｉとを入力変数として、ステップ７００と同じ関数ｆ()を計算し、秘密鍵の保管位置（たとえば、計算機装置２３２が備えるデータベースのレコード番号）を決定し、決定された保管位置から秘密鍵を取得する。

【０１６６】ステップ７２２では、次の版番号ｉ＋１が最大数ｎ以下であるか否かを調べ、ｎ以下の場合には、ｉを１つインクリメントし（ステップ７２４）、ステップ７１４に戻る。一方、次の版番号ｉ＋１が最大数ｎを超えた場合は、エラー処理として終了する（ステップ７３０）。

【０１６７】以上、本発明の第４実施形態について説明した。

【０１６８】なお、上記の各実施形態では、利用者が紛失した秘密鍵を用いて、当該秘密鍵と対の公開鍵でカプセル化されたセッション鍵を開放する場合について説明したが、本発明はこれに限定されるものではなく、利用者が紛失した秘密鍵を用いて、当該秘密鍵と対の公開鍵で暗号化されたデータを復号する場合すべてに適用可能なのはいうまでもない。

【０１６９】最後、以上説明した本発明の各実施形態を、電子商取引システムに適用した場合について説明する。

【０１７０】図１０は、本発明の各実施形態が適用された電子商取引システムの概略構成図である。

【０１７１】このシステムは、図示するように、送信装置８０１と、受信装置８０２と、暗号データ保存装置８１３と、会計監査機関８６１と、鍵回復機関８５１と、送信装置８０１および受信装置８０２間で通信を行うための通信路８００と、で構成される。

【０１７２】鍵回復機関８５１は、図１のＣＡ２２０、ＫＥＣ２３０およびＤＲＣ２４０に相当するものである。受信装置８０２は、図１のＵＳＣ２００に相当する

ものであり、ＵＳＣ２００が搭載する各種プログラム２０２、２０４、２０６、２０８に加えて、暗号データを保存装置８０３に保存するための保存プログラムが搭載される。

【０１７３】送信装置８０１は、取引データを、受信装置２０１の利用者が公開鍵（第１の公開鍵証明書の公開鍵）を用いて暗号化して、受信装置２０１に送信するためのプログラムが搭載されている。会計監査機関８６０は、電子商取引に関して監査手順を実行する。

【０１７４】通常、電子商取引において、暗号化された取引データは、受信後、直ちに復号される。ただし、図１０に示す例では、受信装置８０１は、送信装置８０２から送られてきた取引データを受け取ると、保存プログラムを実行して、たとえば、受信装置８０１の処理能力や作業進行に応じて、受け取った取引データを日／月単位で保存装置８０３に保管するようにしている。そして、後から、受信装置２０１の利用者の秘密鍵（第１の公開鍵証明書の秘密鍵）を用いて復号化することができるようにしている。

【０１７５】受信装置２０１の利用者が秘密鍵を紛失した場合は、受信装置２０１および鍵回復機関８５１間において、上記説明した各実施形態を実施することで、取引データの復号を行うことが可能となる。

【０１７６】なお、会計監査機関８６０は、保存装置８０３に暗号化された状態の取引データを用いて、監査手順を実施することも可能である。この場合、暗号化された状態の取引データを復号するための鍵の入手は、鍵回復機関８５１に依頼する。

【０１７７】

【発明の効果】以上説明したように、本発明によれば、公開鍵証明書および鍵回復における利用者管理を１元化し、公開鍵証明書の発行を受けた利用者に対してのみ、当該公開鍵証明書の公開鍵と対の秘密鍵を紛失した場合に、当該秘密鍵を用いたデータ回復を認めることができる。

【図面の簡単な説明】

【図１】本発明の第１実施形態が適用されたシステムの概略構成図である。

【図２】本発明の第１実施形態において、第１の公開鍵証明書を発行してから、当該証明書を廃止して第２の公開鍵証明書を再発行するまでの処理手順を説明するためのフロー図である。

【図３】図２のステップ１３０（公開鍵証明書の更新手順）におけるデータの流れを概念的に示した図である。

【図４】本発明の第１実施形態において、利用者を認証してから、廃止した第１の公開鍵証明書の公開鍵でカプセル化されたセッション鍵をカプセル開放するまでの処理手順を説明するためのフロー図である。

【図５】図４に示すフローのステップ１４４～ステップ１７０（鍵回復手順）におけるデータの流れを概念的に示した図である。

【図６】本発明の第２実施形態における、図２のステップ１３５に対応するステップ１３５ａでの処理を説明するためのフロー図である。

【図７】本発明の第２実施形態における、図４のステップ１５２に対応するステップ１５２ａでの処理を説明するためのフロー図である。

【図８】本発明の第３実施形態における、図４のステップ１５０に対応するステップ１５０ａでの処理を説明するためのフロー図である。

【図９】本発明の第４実施形態における、図４のステップ１５２に対応するステップ１５２ｂでの処理を説明するためのフロー図である。

【図１０】本発明の各実施形態が適用された電子商取引システムの概略構成図である。

【符号の説明】

２００　ＵＳＣ
２０１、２２１、２３１、２３２、２４１　計算機装置
２０２　電子署名プログラム
２０４　暗号化／復号プログラム
２０６、２２８　公開鍵証明書申請・更新プログラム
２０８、２４３　暗号データ鍵回復・再暗号化プログラム
２２２　データベース
２２６　利用者認証プログラム
２３０　ＫＥＣ
２３０ａ　秘密鍵登録装置
２３０ｂ　鍵保管装置
２３４　鍵供託情報テーブル
２３５　秘密鍵供託プログラム
２３６　鍵供託情報テーブル参照・登録・更新プログラム
２３７　秘密鍵検索プログラム
２４２　回復者認証プログラム
８００　通信路
８０１　送信装置
８０２　受信装置
８０３　保存装置
８５１　鍵回復期間
８６１　会計監査機関

【図１】

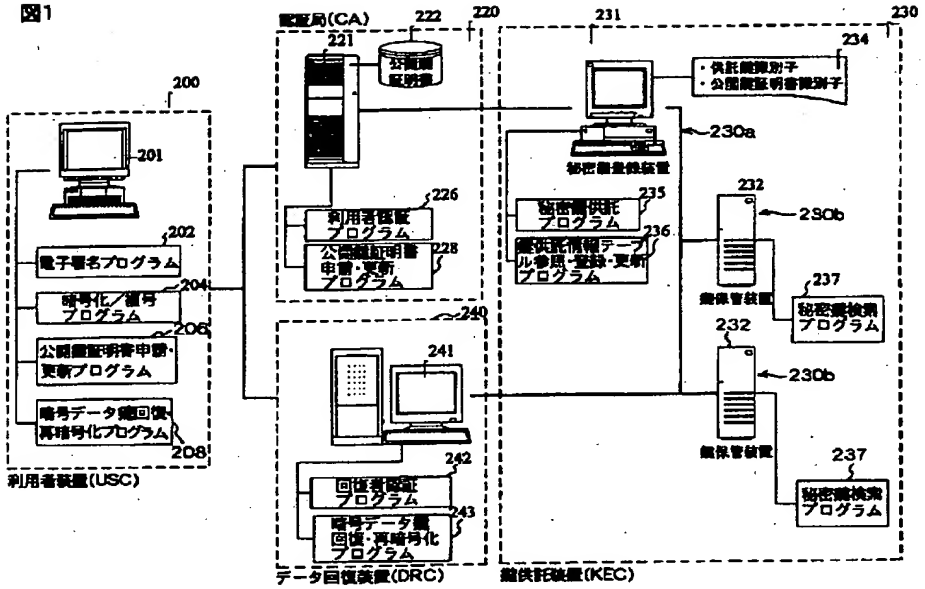


図1
認証局(CA)
222
220
231
230
221
公開鍵証明書
234
・秘託鍵識別子
・公開鍵証明書識別子
200
201
230a
秘密鍵登録装置
226
利用者証明プログラム
232
235
秘密鍵秘託プログラム
230b
202
236
鍵秘託情報テーブル参照・登録・更新プログラム
228
公開鍵証明書申請・更新プログラム
237
秘密鍵検索プログラム
204
暗号化／復号プログラム
206
公開鍵証明書申請・更新プログラム
240
241
232
230b
暗号データ鍵回復・再暗号化プログラム
208
242
回復者証明プログラム
237
243
暗号データ鍵回復・再暗号化プログラム
利用者装置(USC)
データ回復装置(DRC)
鍵秘託装置(KEC)

【図２】

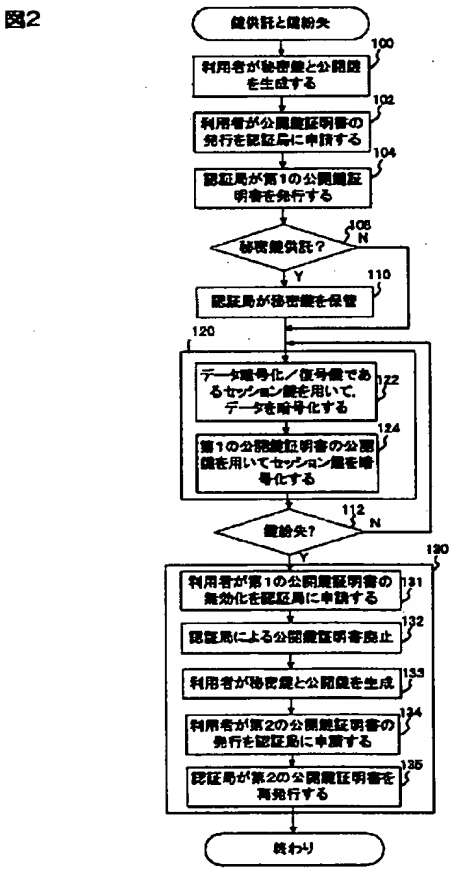


図2
鍵秘託と鍵紛失
100
利用者が秘密鍵と公開鍵を生成する
102
利用者が公開鍵証明書の発行を認証局に申請する
104
認証局が第1の公開鍵証明書を発行する
105
秘密鍵秘託?
N
Y
110
認証局が秘密鍵を保管
120
122
データ暗号化／復号鍵であるセッション鍵を用いて、データを暗号化する
124
第1の公開鍵証明書の公開鍵を用いてセッション鍵を暗号化する
112
鍵紛失?
N
Y
130
131
利用者が第1の公開鍵証明書の無効化を認証局に申請する
132
認証局による公開鍵証明書廃止
133
利用者が秘密鍵と公開鍵を生成
134
利用者が第2の公開鍵証明書の発行を認証局に申請する
135
認証局が第2の公開鍵証明書を再発行する
終わり

【図３】

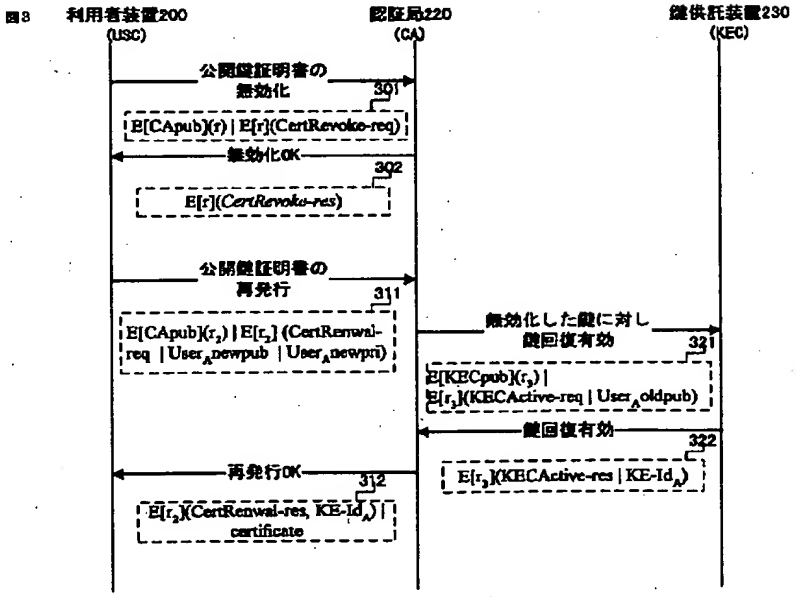
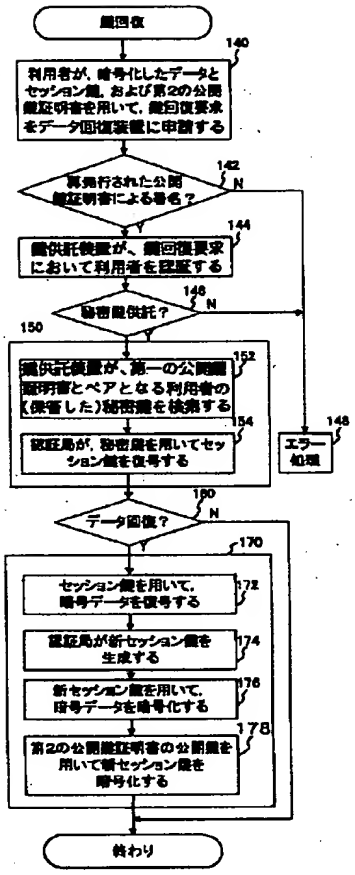


図3
利用者装置200 (USC)
認証局220 (CA)
鍵秘託装置230 (KEC)
公開鍵証明書の無効化
301
E[CApub](r) | E[r](CertRevoke-req)
無効化OK
302
E[r](CertRevoke-res)
公開鍵証明書の再発行
311
E[CApub](r2) | E[r2](CertRenwal-req | User_Anewpub | User_Anewpri)
無効化した鍵に対し鍵回復有効
321
E[KECpub](r3) | E[r3](KECActive-req | User_Aoldpub)
鍵回復有効
322
E[r3](KECActive-res | KE-Id_A)
再発行OK
312
E[r2](CertRenwal-res, KE-Id_A) | certificate

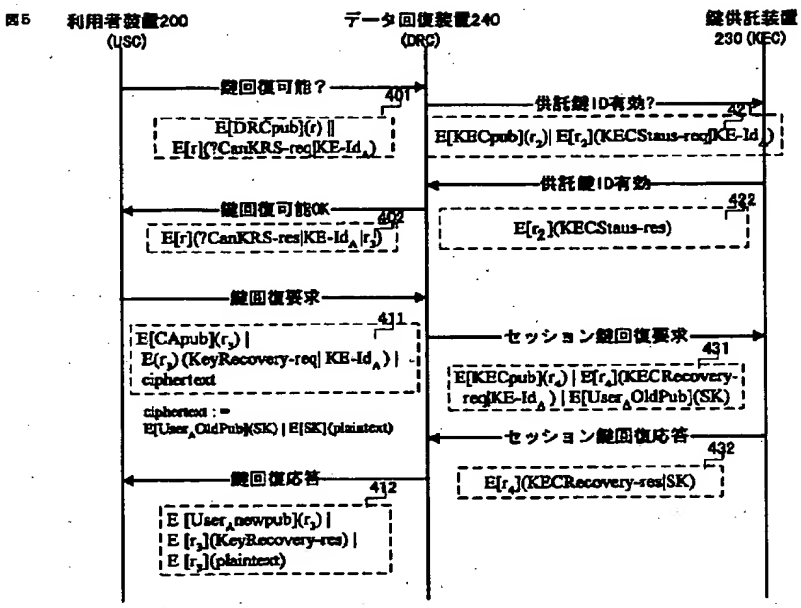
【図４】

図4



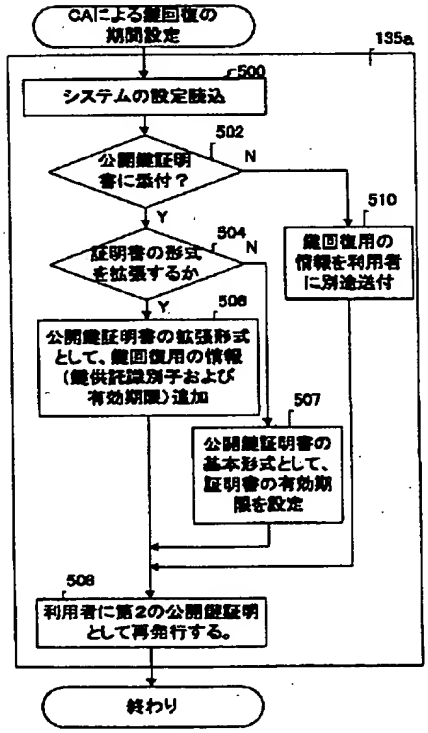

【図５】

図5



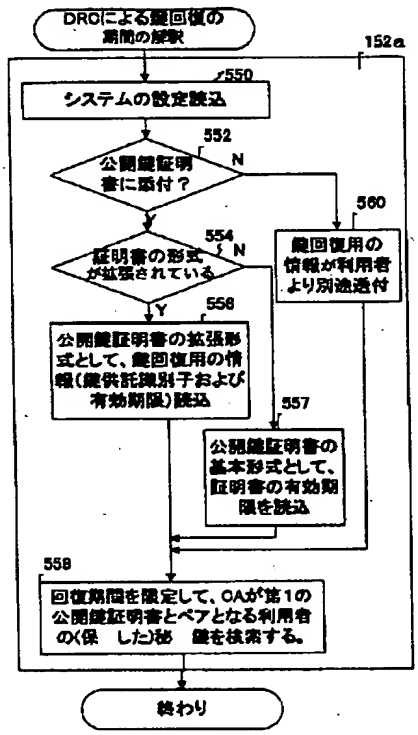

【図６】

図6



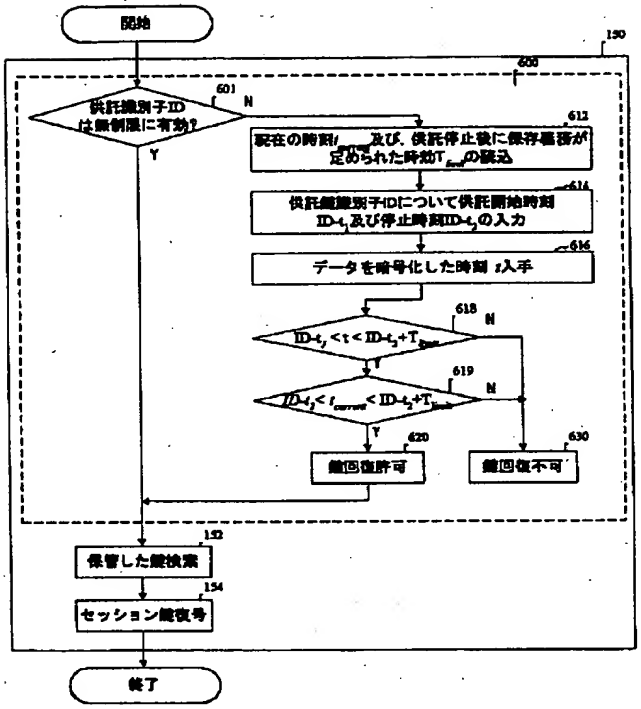

【図７】

図7



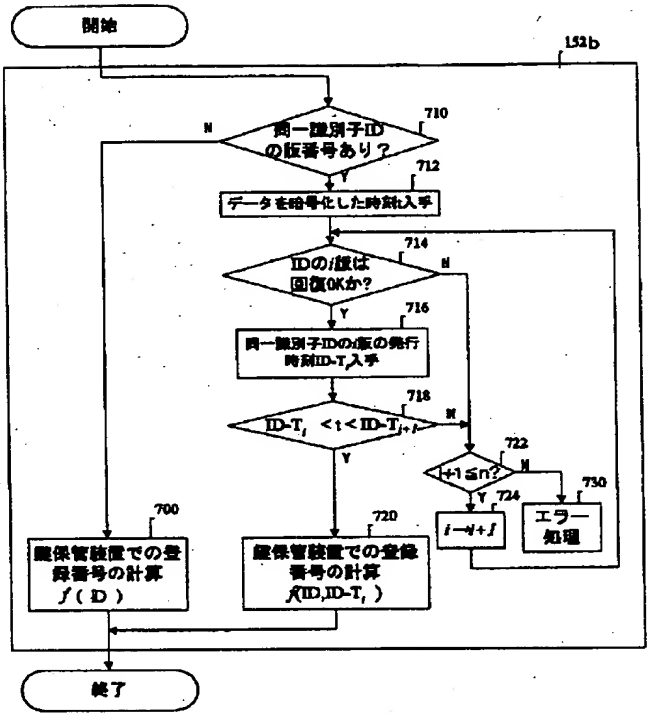
【図８】

図8



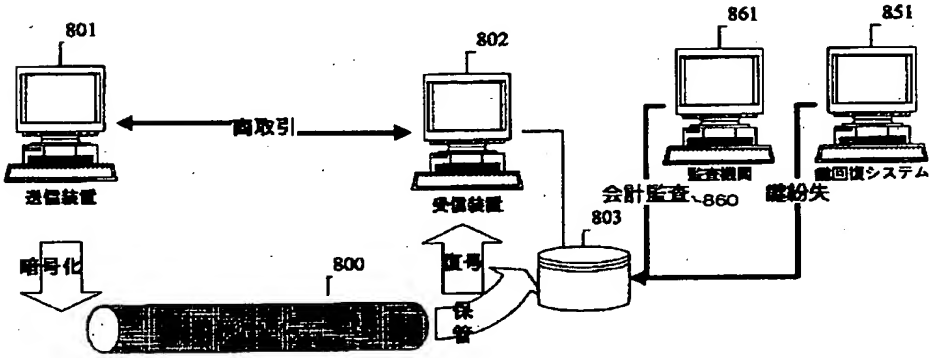

【図９】

図9




【図１０】

図10




---

フロントページの続き

(72)発明者　川井　亨
神奈川県横浜市戸塚区品濃町504番地２
日立電子サービス株式会社内

(72)発明者　柳内　秀敬
神奈川県横浜市戸塚区品濃町504番地２
日立電子サービス株式会社内